Jan-Feb 1978
vol 4, no 1

Austria Sch 43.00
Belgium BF 93.00
Canada $2.50
Denmark Kr 15.00
Finland Mk 9.00
France F 12.00
Germany DM 6.00
Great Britain £ 1.30
Greece Dr 87.00
Holland Df 17.00
Italy L 2000
Norway Kr 13.00
Portugal Esc 73.00
Spain Ptas 160.00
Sweden Kr 11.00
Switzerland SF 6.00
USA $2.00

$2.00

creative computing

the #1 magazine of computer applications and software

Computer Games:
- VAN GAM
- YAHTZEE

A 16-Bit Computer in Your Future?

Murphy's Laws and Computers

Grammar as a Programming Language

How to Write a Computer Simulation

Biorhythm in BASIC and APL

Program for a World Population Model

A New Fast Sorting Algorithm

Robot-Drawing Contest

Blue Wazoo

Your computer system needn't cost a fortune.

Some computer kits include little more than a mother board and a front panel, and you pay extra for everything else you need to make an operating computer.

SWTPC doesn't do it that way, so you can get your Southwest Technical 6800 Computer up and running at a bargain cost compared with most other systems. It comes complete at $395 with features that cost you extra with many other systems.

The Extras You Get

These extras include 4K of random-access memory, a mini-operating system in read-only memory, and a serial control interface. They give you 1) a considerable amount of working memory for your programs, 2) capability through the mini-operating system to simply turn on power and enter programs without having to first load in a bootstrap loader, and 3) an interface for connecting a terminal and beginning to talk with your computer immediately.

Low-Cost Add-Ons

Now that you have a working computer, you'll probably want to add at least two features soon, more memory and interfaces for needed accessory equipment. Memory for our 6800 is another bargain. You can get 4K memory boards for just $100 and 8K boards for only $250.

Our interfaces cost little compared with many other systems. For just $35 you can add either a serial or parallel interface board. (And you won't have to buy several interfaces on a costly board to get just the one you want.)

Peripheral Bargains

Your computer is no good without at least a terminal for entering data and viewing computer output, and you will probably want a good method of storing programs and data.

We offer you a line of high-quality peripherals at low prices. (You can prove this by just comparing prices.)

Buy our CT-64 Video Terminal for only $325 and our CT-VM Monitor with matching cover for $175. Our MF-68 Dual Minifloppy costs just $995, complete with Disk BASIC and a disk operating system. For cassette storage our AC-30 Cassette Interface gives simple control for one or two cassette recorders.

You can get inexpensive hard copy with our PR-40 Alphanumeric Line Printer.

We back up the 6800 system with low-cost software, including 4K and 8K BASIC.

Compare the value you get with our computer and peripherals before you buy. We think you'll find that SWTPC gives you more for your money in every way.

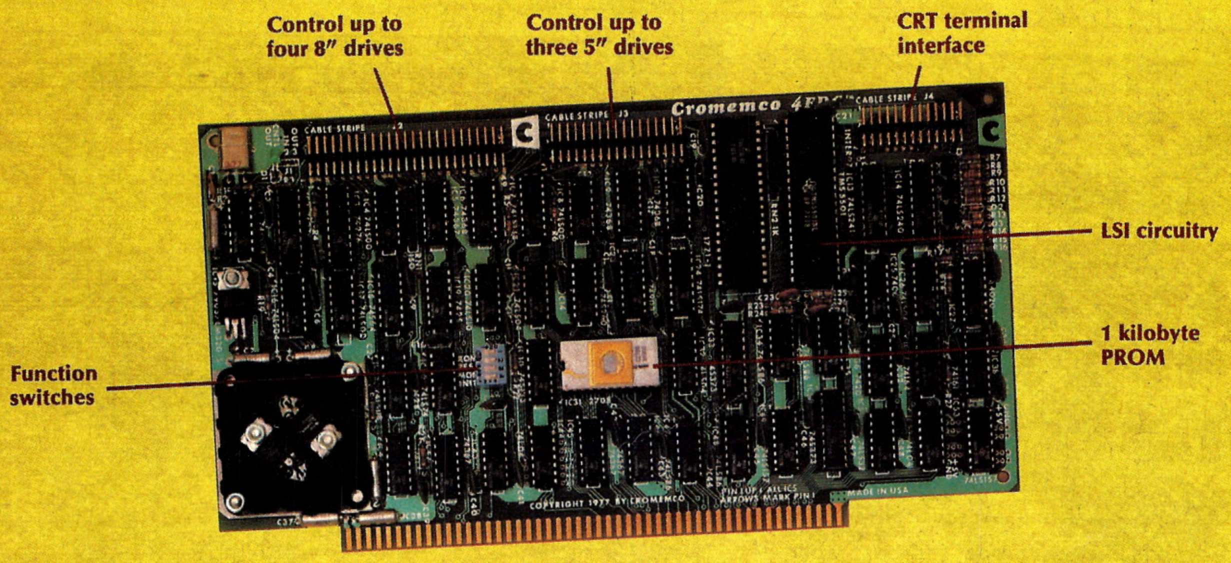
Enclosed is:

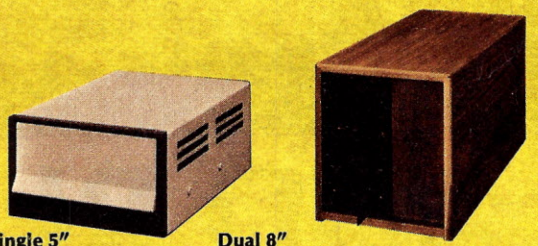
_____ $995 for the Dual Minifloppy
_____ $325 for the CT-64 Terminal
_____ $175 for the CT-VM Monitor
_____ $395 for the 4K 6800 Computer
_____ $250 for the PR-40 Line Printer
_____ $79.50 for AC-30 Cassette Inferface
_____ Or BAC # __________ Exp. Date ______
_____ Or MC # __________ Exp. Date ______

Name ____________________ Address ____________________
City ____________________ State ____________________ Zip __________

Southwest Technical Products Corp.
219 W. Rhapsody, San Antonio, Texas 78216
London: Southwest Technical Products Co., Ltd.
Tokyo: Southwest Technical Products Corp./Japan

The easy way to get disk storage, FORTRAN IV, and other programming power





Single 5″ disk drive

Dual 8″ disk drive

Here's a new disk controller and disk drive combination that will set you up for truly powerful disk storage.

The new controller is extremely versatile. You can use it with **either** our **new 5″ single disk drive** or our **8″ dual disk drive.** In fact, the controller will **interface up to three** 5″ or **four** 8″ drives.

That means you can have enormous disk storage since the new controller puts 92 kilobytes on each side of a 5″ diskette and 256 kilobytes on an 8″ diskette. Recording is in soft-sectored IBM format.

FORTRAN IV AND MORE

You can get still more Cromemco disk operation aids. For example, we also offer FORTRAN IV for our computer users.

And as in so many things, we are the first manufacturer in the field to offer this advanced program for the Z-80 μP.

Besides FORTRAN IV we also offer our special BASIC (14-digit precision), our Z-80 Assembler, and now an entertainment diskette with over a dozen of our Dazzler® games.

KEYBOARD CONTROL

The new Model 4FDC disk controller (supplied in our Z-2D) is for our Z-2 computer or any S-100 bus computer using our Z-80 CPU card.

You should also know about these other capabilities of the new controller:

- Its PROM-resident Disk Operating System (RDOS) gives you keyboard control of your disk drive and also includes a bootstrap to load our powerful CDOS disk operating system supplied on all Cromemco diskettes.
- The controller will **interface your CRT terminal** through its RS-232 serial port. May save you an I/O.
- It has 5 programmable interval timers.
- It has vectored interrupts.
- And it has an 8-bit parallel input port and an 8-bit parallel output port.

LOOK TO THE FUTURE

This new disk controller equips you for the future as well as for now. Not only can you now have very large storage, but the features of the controller and the standard IBM format protect you from early obsolescence.

STORES/FACTORY

This new card and the disk drives are in production and available.

So contact your computer store or the factory today and you can have the power of FORTRAN IV and a large memory right away.

PRICES

Model 4FDC-K Disk Controller kit	$ 395
Model 4FDC-W Disk Controller assembled	$ 595
Model WFD 5″ single disk drive assembled	$ 495
Model PFD-K 8″ dual disk drive kit	$1995
Model PFD-W 8″ dual disk drive assembled	$2495

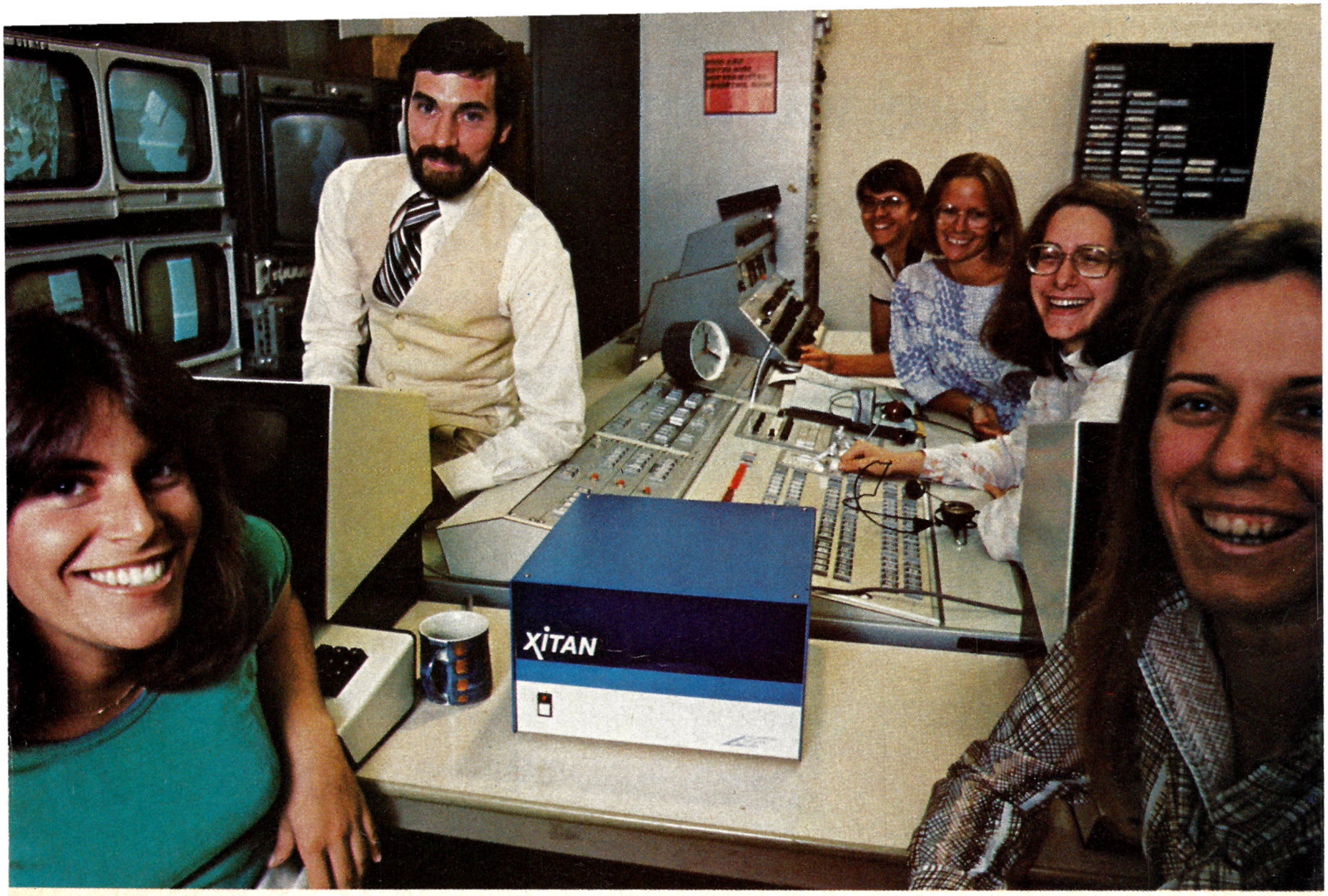
Disk drives are complete with power supply, case and cables.

SOFTWARE

Purchasers of Cromemco computers or drives may purchase software on 5″ or 8″ diskettes as follows:

	5″ Diskette Model	8″ Diskette Model	Price
FORTRAN IV	FDF-S	FDF-L	$95
Z-80 Assembler	FDA-S	FDA-L	$95
16K BASIC	FDB-S	FDB-L	$95
Dazzler® games	FDG-S	FDG-L	$95

Cromemco incorporated
Specialists in computers and peripherals
2400 CHARLESTON RD., MOUNTAIN VIEW, CA 94043 • (415) 964-7400



UP AND RUNNING

TDL EQUIPMENT USED BY NEW JERSEY PUBLIC TELEVISION TO PROCESS NEW JERSEY GUBERNATORIAL PRIMARY ELECTION RETURNS

John Montagna, computer engineer (above left), lead this successful network team in generating election results speedily, efficiently and reliably using predominantly TDL hardware and software. Montagna created three programs to get the job done. The text for a SWAPPER program was written and assembled using the TDL TEXT EDITOR and Z80 RELOCATING MACRO ASSEMBLER. The SWAPPER text and all debugging was run through TDL's ZAPPLE MONITOR. The relocatable object code was punched onto paper tape. A MAIN USERS program updated votes and controlled air display. An ALTERNATE USERS program got hard copy out and votes in. The latter two programs were written in BASIC. Montagna modified the ZAPPLE BASIC to permit time-sharing between the two USERS programs.

Four screens were incorporated, two terminals entered votes as they came in and were used to call back votes to check accuracy. Montagna called on the power and flexibility offered by TDL's ZPU board and three Z-16 Memory boards.

Montagna's setup worked constantly for over four hours updating and displaying state-wide and county-wide results without flaw.

"I chose TDL because they have all the software to support their hardware, and it's good; it has the flexibility to do the job."
John Montagna

We salute John Montagna and NEW JERSEY PUBLIC BROADCASTING for spearheading the micro-computer revolution.

TDL's XITAN SYSTEMS have the capacity to do similar tasks for you. Write to us for XITAN information and the name of your nearest TDL dealer.

TECHNICAL DESIGN LABS

RESEARCH PARK BLDG. H 1101 STATE ROAD
PRINCETON, NEW JERSEY 08540 (609) 921-0321

in this issue . . .

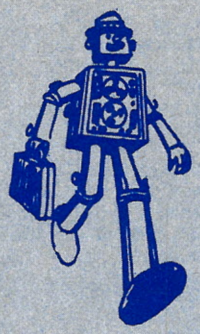
articles

fiction & foolishness

reviews & resources

things to do - games

departments



The Cover

That big bird on the cover is the Blue Wazoo, a steel-and-plastic-and-electronics structure six feet high, with some peculiar behavior patterns. For the full story, see page 30.

Jan-Feb 1978 Volume 4, Number 1 Consecutive Issue No. 19

Creative Computing magazine is published bi-monthly by Creative Computing, P.O. Box 789-M, Morristown, NJ 07960. (Editorial office: 51 Dumont Place, Morristown, NJ 07960. Phone: (201) 540-0445.)

Domestic Subscriptions: 1-year $8, 2-year $15, 3-year $21. Subscription orders, change of address, P.O. Form 3579 to Creative Computing, P.O. Box 789-M, Morristown, NJ 07960. Call 800-631-8112 toll-free to order a subscription (to be charged only to a bank card).

Foreign Subscriptions:
Great Britain. 1-year £ 6.25, 2-year £ 11.90, 3-year £ 16.90. Orders to Creative Computing, 60 Porchester Road, Southampton S02 7JD, England.
Europe, 1-year subscription. Austria Sch 180, Belgium BF 4000, Denmark Kr65, Finland Mr40, France Fr50, Germany DM25, GreeceDr360, Holland Dfl27, Italy L8500, Norway Kr55, Portugal Esc320, Spain Ptas700. Sweden Kr45, Switzerland SF 25. Orders to Pan Atlantic Computer Systems GmbH, Frankfurter Strasse 78, D61 Darmstadt, German Fed Rep.
Other Countries 1-year $12, 2-year $23, 3-year $33 (surface postage, U.S. dollars). Orders to Creative Computing, P.O. Box 789-M, Morristown, NJ 07960.

Second class postage paid at Morristown, New Jersey and at additional mailing offices.

...notices...

Intel University Software Contest

More than $28,000 worth of microcomputer development equipment, plus numerous $100 memberships in the Intel User's Library, will be awarded by Intel Corporation to colleges and universities in the United States and Canada.

The equipment will be awarded as prizes during a University Software Contest that runs through **June 30, 1978.** A team of software specialists will evaluate software programs for their originality, documentation, creativity and applicability to microprocessors. Students and faculty members can enter as individuals or teams. Prizes will be awarded to the winning schools.

Intel's Software Library, Insite, is encouraging the contribution of original, high-quality programs. Examples include operating systems, communications programs, scientific subroutines, language translators, programmable peripheral-device subroutines, and applications programs in the areas of numerical control, data acquisition, process control, data analysis and lab automation.

Entries must be written in Intel assembly language or PL/M and must include a source listing and test program to assure program validity and a source paper tape or diskette. Entry forms and contest details may be obtained by writing: Insite Library Contest, Intel Corporation, Microcomputer Division, 3065 Bowers Ave., Santa Clara, CA 95051.

NCC '78

Billed as "the year's largest annual gathering of the computing and data processing fields," the 1978 National Conference will be held **June 5-8, 1978,** in the Anaheim Convention Center, Anaheim, California.

NCC '78 will feature about 100 technical sessions, a series of professional development tutorials, exhibits of computer products and services by about 300 organizations, plus special addresses, events, and activities.

A special feature of NCC '78 is the Personal Computing Festival, to take place **June 6-8** at the Disneyland Hotel complex in Anaheim. Some 30 sessions are planned, and prizes will be awarded for the most innovative demonstrations of individually-designed hardware and software systems and applications.

AFIPS Headquarters, 210 Summit Ave., Montvale, NJ 07645. (201) 391-9810.

PERCOMP '78

Co-sponsored by the International Computer Society/SCCS and the Rockwell Hobbyist Computer Club, PERCOMP '78 will be held at the Long Beach Convention Center, Long Beach, California, **April 28-30, 1978.** The show is "designed with the home computerist and small-business person in mind."

James Lindwedel, Technical Program Chairperson, PERCOMP '78, 1833 E. 17 St., Santa Ana, CA 92701.

Your Key to Fame and Fortune

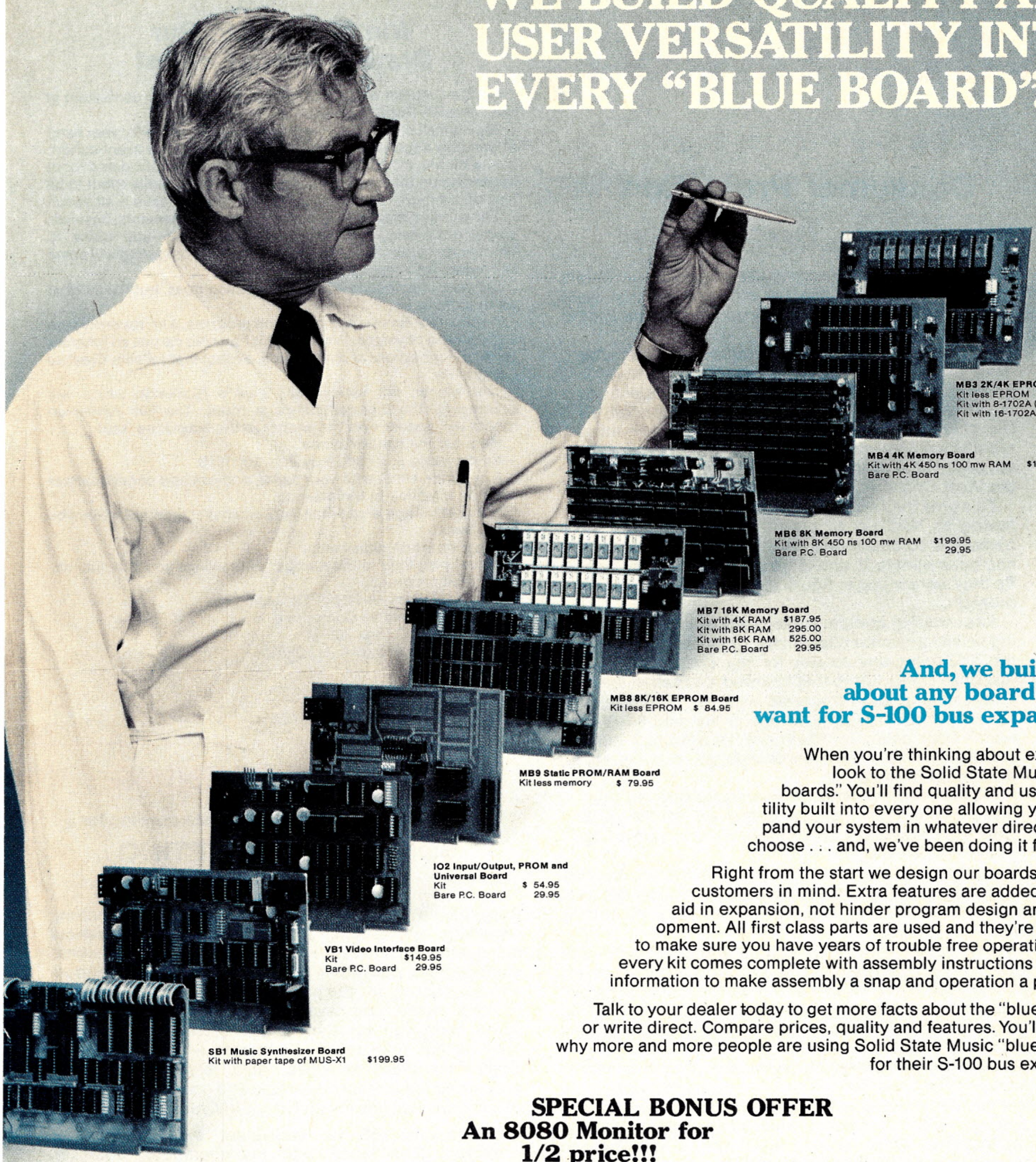
Creative Computing is seeking "how-to" articles on several specific topics for upcoming issues. The areas of interest are:

1. **Investment portfolio analysis.** For example, comparative stock analysis, stocks vs bonds vs treasury bills vs other alternatives, puts and calls, arbitrage analysis, etc.
2. **CAI author system.** Software system that allows the user to write a tutorial dialog of questions and answers for any subject. PLANIT is an example of such a system for large computers; HP has an Instructional Dialog Facility for their 2000 and 3000 series: DEC has DECAL, a relatively unsophisticated system that runs under RSTS. We are seeking to publish a system for a microcomputer system with floppy disk.
3. **Sports Smiulations.** In-depth, serious simulations of various sports such as ice hocky, soccer, gliding, sailing, etc. The idea is that the computer player can learn something about playing tactics and strategy that can be used in the actual sport.
4. **Speech recognition and synthesis.** Perhaps an interactive version of ELIZA (Jul/Aug 1977, pg 100) or some other interesting program.

Articles should be truly of a "how-to" nature, not speculations or philosophy. We're looking for software with listings and sample output and a complete description of your algorithms and approach. Should be typed, double-spaced. Output with clean black ribbon on white paper. *Payment upon acceptance* (not upon publication) at our usual rates. Which is to say, you'll not get rich but you'll be compensated for your time and effort. Photos and/or illustrations are highly desirable and will be paid for also if they're suitable for publication.

If you wish your material returned in the event it is not accepted for publication, please include a self-addressed stamped envelope (SASE) of adequate size and enough postage for its return. We cannot guarantee acknowledgment of contributions without an SASE.

WE BUILD QUALITY AND USER VERSATILITY INTO EVERY "BLUE BOARD"...



ssembled boards also available t slightly higher prices.

And, we build just about any board you'll want for S-100 bus expansion.

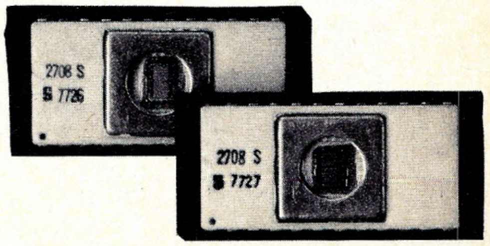
When you're thinking about expansion look to the Solid State Music "blue boards." You'll find quality and user versatility built into every one allowing you to expand your system in whatever direction you choose . . . and, we've been doing it for years.

Right from the start we design our boards with our customers in mind. Extra features are added that will aid in expansion, not hinder program design and development. All first class parts are used and they're checked to make sure you have years of trouble free operation. Plus, every kit comes complete with assembly instructions and user information to make assembly a snap and operation a pleasure.

Talk to your dealer today to get more facts about the "blue boards" or write direct. Compare prices, quality and features. You'll find out why more and more people are using Solid State Music "blue boards" for their S-100 bus expansion.

SPECIAL BONUS OFFER
An 8080 Monitor for 1/2 price!!!

If you buy any of the Solid State Music kits or assembled boards you'll receive a SSM8080 Monitor complete with either eight 1702's or two 2708's and over 50 pages of software information. A $49.95 retail value . . . just $25.00. Hurry!



cybercom

Solid State Music
2102A Walsh Avenue
Santa Clara, CA 95050
(408) 246-2707

We're the blue boards

.editorial..

Consumer Protection

No doubt all the hobby-computer magazines get letters from readers such as one we recently got that went something like this (heavily paraphrased so the sender won't recognize it):

> A year ago I ordered an XYZ computer kit, and after three months went by, and I hadn't heard from the XYZ Company, I called them to find out what happened to my order. They couldn't find it. I stopped payment on the check, and sent another order. Another three months went by without a word from the XYZ people, so I called again. (I'm on the East Coast, and they're on the West Coast, so these calls aren't cheap.)
>
> This time they said they had my order, but they were having to hold up all orders for the computer kit because a company making one of the important ICs had gone bankrupt, and there was no second source. So they were trying to figure out some way around he problem.
>
> After waiting another three months, and discovering newer and more interesting computer kits, I wrote the XYZ company to ask for my check back. No response in the month that followed, so I called them. They said they hadn't received my letter. So I asked to have the check returned to me. They said sorry, it wasn't their policy to return checks without a letter requesting this. I sent another letter, and am still waiting for my refund.
>
> Please print my letter in your magazine, so your readers will know what's going on at the XYZ Company, and why I would never try to buy anything from them, ever again."

We get that type of letter every now and then, and the magazines that specialize in hardware probably get a lot more of them. The question is, should we print such a letter?

We've talked with other editors and publishers, and found varying viewpoints. One said "If a company is just getting off the ground, why give them the kiss of death just because they're having growing pains?" Another felt that "We've got to be careful, because we can't afford to lose any advertising."

Most of us have stayed away from this ticklish subject of printing "I've been had" letters, but we can't ignore them forever. We're going to take a much closer look at letters dealing with problems involving software and applications, which is our area, and if we find the complaint is valid, then we should let you, our readers, know that the ABC company doesn't deliver software as promised, or that the DEF Corporation's inventory program doesn't live up to spec, etc.

After all, how long are growing pains supposed to last?

— *Stephen B. Gray*

Win $25-Plus In Our Draw-A-Robot Contest

Be the *Creative* winner, or at least be a *Creative* contestant in our "Draw-A-Robot Contest."

We want all of you *Creative Computing* readers to try your hand at creating a robot. Not a mechanical demon or a lively cohort. Just a simple, modern, humanized version of our out-dated robot appearing on the contents page. We need a versatile robot to be part of our logo—one that symbolizes what *Creative* is all about.

The winning robot will appear on our advertisements, business cards, stationery, stickers, in the magazine—you name it. Whomever comes in contact with *Creative Computing* will come in contact with *the* robot.

Let your imagination run wild. Anything goes, but here's what we're looking for....

A robot that reveals character and progress, a trustworthy face and a strong physique. A robot in motion who carries an inherent sense of innovation. In other words, a robot that reflects *Creative Computing.*

The winner will receive $25 plus a biosketch which will accompany the featured full-page robot. The five runner-up robots will appear with the winner in the May-June issue.

Here are the specifications:

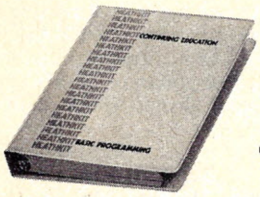
—All drawings must be in by Feb. 20, 1978
—A separate sheet with your name, address and phone number must be attached to the drawing.
—Robot drawings larger than 9x12 or smaller than 1x3 will not be considered.
—Black on white paper only.
—There is no limit to the number of drawings you can submit.
—Send all robots to:

Draw-A-Robot Contest
Creative Computing
P.O. Box 789-M
Morristown, NJ 07960

Heathkit "Program-Ready" Computer Systems Await YOUR COMMANDS!

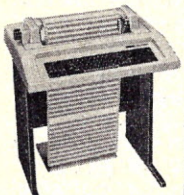
The Heathkit H8 is designed to provide you with a versatile, efficient computer that's easy and fun to build and use. It features an intelligent front panel with keyboard entry and 9-digit display, a heavy-duty power supply with enough extra capacity for memory and I/O expansion, and an exclusive 50-line fully buffered bus capable of addressing 65K bytes. The CPU board is fully wired and tested and features the 8080A chip, clock, systems controller and full bus buffering. Seven vectored interrupts are available on the bus for quick response to your I/O requests. The mother board has positions for up to 10 plug-in circuit boards for memory and interface cards and complete "program ready" software (BASIC, assembler, editor and debug) is included at no extra cost. Together with its "system designed" peripherals: memory expansion boards, serial and parallel interfaces, the H9 CRT terminal with ASCII 67-key keyboard and the ECP-3801 cassette recorder/player mass storage device, the H8 will provide you with years of computing usefulness and enjoyment.

Based on the world-famous DEC LSI-11, the H11 and its peripherals give you all the power and speed you need for total computing versatility. It's one of the few FULL 16-bit computers available to the hobbyist today, and equivalent commercial versions would cost literally thousands of dollars more. The H11 features a fully wired and tested DEC KD11F board that contains the 16-bit LSI-11 CPU, a 4096x16 read/write MOS semiconductor memory, DMA operation, and includes the powerful PDP-11/40 instruction set. Heath/DEC PDP-11 software includes editor, relocatable assembler, link editor, absolute loader, debug, I/O executive and DUMP programs, plus BASIC and FOCAL. The backplane/card guides are fully compatible with all standard LSI-11 accessories. The H11 and its system peripherals: memory expansion modules, serial and parallel interfaces, an extended arithmetic chip, plus the H10 paper tape reader/punch and the H9 CRT terminal give you all the power and versatility you need for personal computing at its very best!



Basic Self-Instruction Course

$29.95 Learn BASIC programming easy and fast with our EC-1100 self-instructional course. Features programmed instruction text, experiments and demonstrations to give you a thorough knowledge of BASIC and programming techniques.



World-Famous DECwriter II

$1495 Deluxe keyboard printing terminal at Heath's sensational low mail order price! Also available: LAXX-KG EIA (RS-232C) Interface with cable and connector, **$65**; LAXX-LM Acoustic Coupler which mounts in LA36 with hardware provided, **$250.00**

FREE!

HEATHKIT CATALOG

The new Winter '78 Edition

You can get a FREE retail catalog by redeeming this coupon in person at any of the 50 Heathkit Electronic Centers (Units of Schlumberger Products Corporation) in major markets coast-to-coast, where Heathkit products are sold, displayed, and serviced. (Retail prices on some products may be slightly higher.) (Check the white pages of your telephone book.)

Use coupon for mail-order catalog or bring to store for retail catalog.

Nearly 400 fascinating electronic kits fully illustrated and described, plus a variety of Heath-recommended assembled products. Send for your *FREE* copy today!

Prices are mail order net FOB, Benton Harbor, Michigan. Prices and specifications subject to change without notice.

HEATH
Schlumberger

Heath Company, Dept. 355-370
Benton Harbor, Michigan 49022

Please send me my FREE Heathkit Catalog.
I am not on your mailing list.

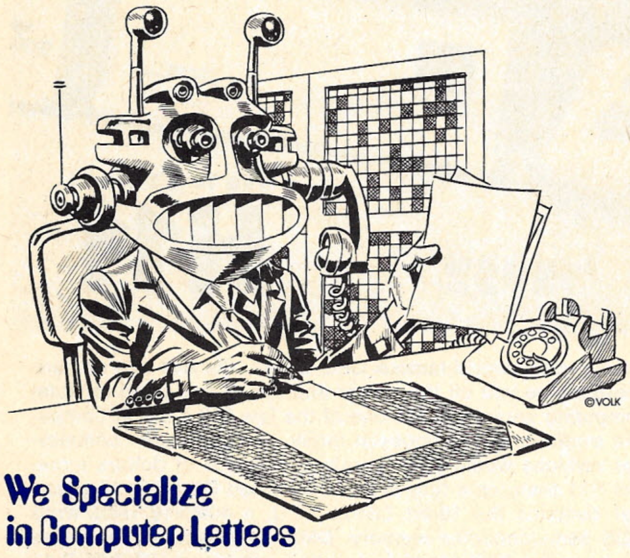
Name______________________________

Address____________________________

City____________________ State__________

CP-137 Zip__________

put...input/output...in



Pen Pals for PLATO and MULTITUTOR?

Dear Editor:

I am very interested in Computer *Assisted* Education (emphasis on assisted) and believe that it will be, and now is to some extent, a great aid to educators. I would like to correspond with others who also have an interest in this subject. I am especially interested in the PLATO and MULTITUTOR systems, both of which use the TUTOR language. A computer should never take over completely the role of educator, but systems like PLATO encourage and assist out-of-class and extra study and also stimulate interest. They make the job of the educator easier and give him/her more time for individual interaction with the students. I believe that this makes for a closer relationship between teacher and student and creates an atmosphere that is more conducive to learning, even in today's very large college and high school classes.

I am also glad to see that your magazine publishes computer games. Many computer "purists" (as they call themselves; I call them other things) believe that such games are below them and are childish. I believe that they are exactly what they are meant to be; fun, recreational, and sometimes even educational. I have the book *101 BASIC Computer Games,* edited by your publisher David H. Ahl, and both volumes of your *Best Of Creative Computing* and enjoy all three. I have, in total, about 400 games recorded on a 2400-ft. tape and on listings. I would enjoy corresponding and trading programs with others who enjoy computer games. The games I have range from very simple number-guessing games to very complex science fiction simulations. I hope that you continue to give games a place in your magazine.

I wish you continued luck with your magazine and hope that more people will take the attitude that computers can be fun.

Bruce A. Carter
7786 South 300 West
Union Mills, IN 46382

A Fan Tells All

Dear Editor:

From the beginning of '77, CC seems to have been rolling. I remember a letter that I got shortly before my first issue back in Nov '75 saying that CC needed subscribers badly so that CC could increase the size of the magazine, print on better quality paper, increase the issue frequency, and to let the staff buy groceries. Well, with the advent of the rise in microcomputer sales to hobbyists, most of that has come true. Tell me, does the staff buy groceries?

From Tom Allen's "Algorithmic Basic," I see that Tom is hooked on COBOL. My advice to him is to learn ALGOL and try pseudo-ALGOL Algorithmic Basic. After all, ALGOL is for the algorithms, not COBOL.

As Tom pointed out, picking apart algorithms is difficult and time-consuming. This one Star Trek program was a doo-zee for me. If only the programmer would take the time to document his programs and explain the complex algorithms, rather than cramming it all into a few lines, the problem would disappear.

Two particular evils are multiple statements on one line, usually separated by colons (:), and lack of spaces. For example, the statement 10READY might appear to the unaware person as a statement called READY rather than READ Y. Two examples of these evils are POSTER in Nov-Dec '76 which uses so many multiple statements that it boggles the mind, and TICKERTAPE in May-Jun '77 which also uses a third evil known as three-letter abbreviations.

I would like to thank Gregory Yob for his excellent article on PILOT. I am currently writing a PILOT interpreter for an HP 2000F computer and one for a CDC Cyber 73-26 KRONOS system which has two CPUs.

A special thanks to CC and Steve North, Jeff Shrager and the creator, Joseph Weizenbaum for bringing a *Basic* version of ELIZA to CC. I think it makes an interesting demonstration and it's good for just plain fun. I have a partially completed HP 2000F Basic version, which, by the time you read this, should be completed.

I would appreciate receiving letters from people interested in computing, especially APL fanatics, people who use PILOT, Artificial Intelligence programs (like ELIZA), and especially people that make, or at least use, computer games in BASIC. I have about ten programs that I would like to share with interested people. Thanks a lot.

Steve Trapp
5020 Mulcare Drive
Columbia Heights, MN 55421

Calculating the Number of Coconuts

Dear Editor:

This is in regard to the Sailor program in the May-June 1977 "Thinking Strategies" (p 78). In case no one else has sent it in, there is a much more efficient algorithm for calculating the number of coconuts, that does not require looping, repeated incrementing, nor conditional tests (except one): If "x" is the smallest possible number of coconuts, and "n" is the number of sailors, then:

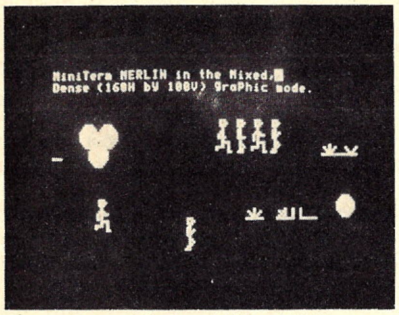
For even "n"; $x=n^{(n+1)} - n^n - n + 1$

For odd "n"; $x=n^n - n + 1$

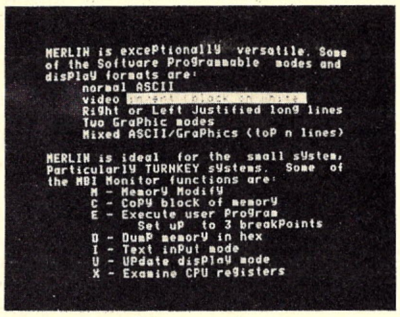
This gives the correct result for any "n" greater than 2. This algorithm was given in *65 Notes,* Vol. 2, No. 1, page 9 (January 1975). *65 Notes* is the publication of the HP-65 Users Club.

Delmer D. Hinrichs
2116 S. E. 377th Ave.
Washougal, WA 98671

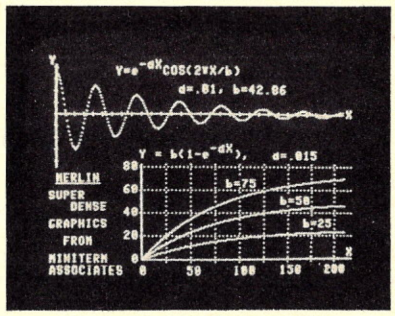
The many faces of MERLIN



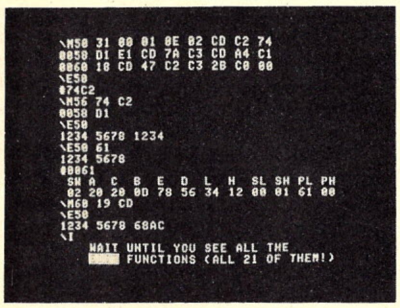
Dense Mode: 160H × 100V
Running Man Patterns



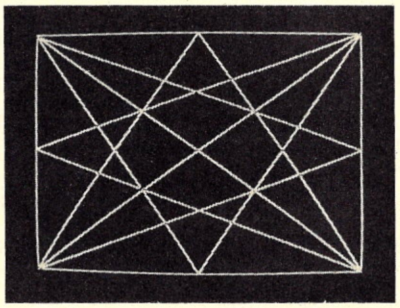
Propaganda



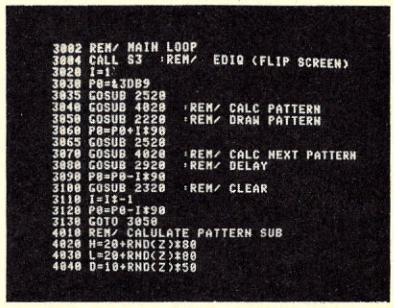
Super Dense: 320H × 200V
Equation Plotting



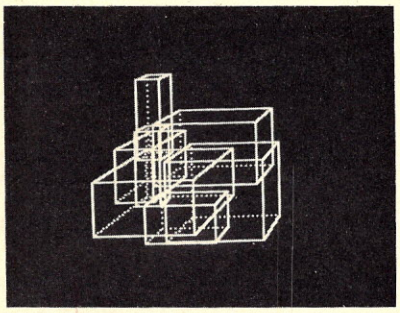
Monitor Debug Usage



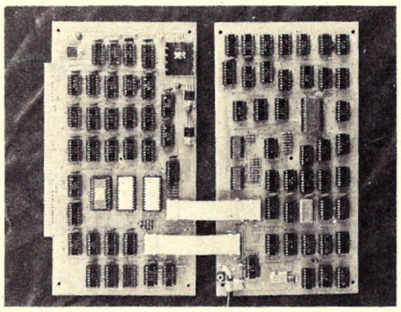
Super Dense: 320H × 200V
Line Drawing

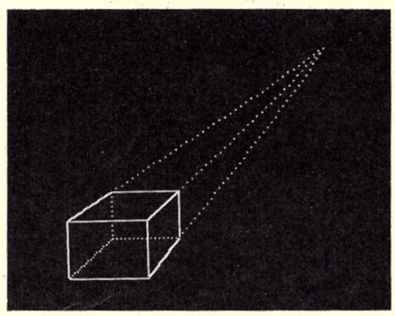


BASIC Program Listing
Output Shown Below



Super Dense: 320H × 200V
3-D Boxes





Super Dense: 320H × 200V
Perspective Drawing

MERLIN (and your S-100 Computer...)
... the graphics development package you've been waiting for.

Do these photos suggest an application in your field? Whether you're into architecture, astrology, music or whatever, you can apply graphics to make your presentations more effective and your work more efficient.

Real Time Plotting
- Heart Rate
- Navigation
- Spectrum Analysis

Complex Equation Plotting
- Stock Market Trends
- Teaching: Mathematics
- Circuit Responses

Fine Line Drawings
- Architecture
- 3-D Projections
- Circuit Layouts

Pattern Movement
- Animation
- Games

MERLIN is also a multipurpose system monitor board with ROM monitor and editor software, parallel keyboard port, and audio cassetts storage, besides being your text and graphics video output device.

Write or call today for your free copy of our new catalog, an index to available application notes, and a list of **MiniTerm** Dealers.

MERLIN is supported by expert technical assistance, a constantly expanding series of application notes, and a newly formed users group.

Assembled and tested **MERLINs** start at $349, $269 for kits. Super Dense add-on, firmware, and cassette interface are extra.

MiniTerm Associates, inc.

Dundee Park, Andover, MA 01810 (617) 470-0525

put...input/output...in

File Structures Revisited

Dear Editor:

Pertaining to the "Reader Challenge" on page 32 of the Nov-Dec issue, your specifications for the LP IS&R system are probably good enough if you are just maintaining a record-store-type inventory. However, for a collector of records, it just doesn't contain enough information. My collection is heavy on Dixieland records, and if all I could record in my system was the Schwann-catalog-type information, it would not be of any real use to me. For example, in addition to wanting to know the LP title & record number, I also need to know the title of all the individual selections on the LP, I would also like to know the artists who recorded that selection, the date it was recorded and the original label and record number (if available). I am going to want to know, for example, how many versions I have of "Muskrat Ramble," or I may want to know if I have a version of "Muskrat Ramble" featuring Kid Ory, or I may want to know (especially when buying additional records) if I already have these selections on another LP.

You might also want to throw in such information as: mono, stereo, quad, etc; the recording characteristics; the sequential arrangement of multi-record sets (i.e., in a three record set is the record/side arrangement 1-6/2-5/3-4 or is it 1-2/3-4/5-6); the condition of the record, the number of times played; etc.

These would seem to be some of the features that one would want to build into the system to offer greater utility to the user. After all, I can now maintain a paper record catalog of my collection. But by expanding the information available I can make the file more useful. I can look for a particular artist doing a particular selection, for example. If I add a "play count" I can see which records are starting to wear out and I can either put them on tape or I can shop for another LP.

John Lees' article on "Programming Techniques: File Structures (Part 1)" looks like it is going to be a useful series. However, it might have been more useful if a small change had been made in the "formats" for the 0, 1 & 2 type records. You can make your retrievals much easier if he had specified the record formats as:

Type 0 : Author : Title : P-Code : Binding : Pages : Price : Date
Type 1 : Author : Title : P-Code : Binding : B-Code
Type 2 : Author : Title : B-Code

or even better:

Type 0 : B-Code : Author : Title : etc
Type 1 : B-Code : Author : Title : etc
Type 2 : B-Code : Author : Title : etc.

In the first example you would be able to say go to field two/three and look for author and/or title, or vice versa. However, you would probably have problems if you ever want to do something like listing your collection in sort on author/book title/story title, as the only thing you have to tie stories into a book collection title is the B-code and its location is not uniform.

In that case, if you go to the second set of examples, you have the author, title, and a method of tying collections together via the B-Code all in the same fields, which makes your retrievals/sorts easier. Having raised this point, I now wonder how I am going to solve the question of how do you handle a collection of short stories by different authors? You have the editor showing up as the author of the book, and the actual author showing up as the author of the short story. Well, your readers can figure that one out for themselves.

In this particular case study, the author might want to use the IBSN as a replacement for the P-Code and the B-Code. The IBSN is the International Book Serial Number, which uniquely identifies a book as to publisher and title.

David Williams
5079 Blacksmith Drive
Columbia, MD 21044

Reply to "An Amazing Computer System"

Dear Editor;

In the July-Aug 1977 *Creative Computing* there was a letter from a poor soul who thought that an IBM 1130 was a fantastic deal. Well, I suppose that if one wants a machine which has the letters 'I' 'B' and 'M' on its panel, and 1130 would be a good deal.

However, I feel that any sixteen-bit processor (i.e., Digital Equipment Corp.'s LSI-11, Data General's Nova, etc., with approx. $65-68,000 worth of peripherals would be a much better match of processor to peripherals. Even this type of a match is a bit ridiculous.

For $70,000, a more realistic system would timeshare, have possibly two disk cartridge drives, some sort of tape-storage media, most likely a sixteen-bit processor, 64K of memory (or 128K bytes equivalent as Mr. Berlin says) and say 6-8 terminals.

A system such as this can be purchased from any vendor (excepting, of course, IBM) for the range of $60-80,000. May I suggest that any installation thinking of buying a single-user or batch-oriented system think twice, for the last of the great batch-machine manufacturers (IBM) probably wishes that it could get out of batch into timesharing without lawsuits and other legal problems. Of course I am not talking about totally getting rid of batch processing; it still has very many valid uses. But for a school, a total batch system for $70,000 dollars is a waste of money, especially one with as sickly a processor as an 1130.

I suggest to the owners (leasers) of such a system: junk it and get a real machine.

IBM Hater

P.S. Keep up the interesting work with the magazine.
P.S.S. Pardon the illiteracy in the preceeding letter.

Finding the Day of the Week

Dear Editor:

In your issue of Nov-Dec 1976, James Reagan described Zeller's congruence, a formula for finding the day of the week given the numeric month, day of the month and year. The formula, as amended by Michael Smith in the Mar-Apr 1977 issue is:

F=(INT(2.6*M-0.2)+K+D+INT (D/4)+INT(C/4)-2*C)mod7

where M=numeric month-2 (January & February become 11 and 12 and the year is reduced by 1)

K= day of the month
D= last two digits of year

C= first two digits of year.

I recently had occasion to make use of Zeller's Congruence in a computer program and found an error, not in the expressed equation, but in its application to computers using binary or hexadecimal floating-point notation. The error occurs in the first part of the equation; INT(2.6*M-0.2). When the month is February, M=12 and INT(2.6*12-0.2)=31, on paper. On the computer, the result is 30, which messes up the entire month of February. This is because of the peculiar fact that numbers like .2 and 2.6 cannot be expressed exactly in binary, but are repeating binary fractions. The number .2 converts to the binary fraction .001100110011, those twelve digits repeated infinitely. Consequently, instead of 2.6*12-0.2, the computer uses something like 2.599999*12-0.199999=30.999989, whose integer portion is 30, not 31. This problem is readily corrected either by using INT (2.6*M-0.1999) or the preferable INT((26*M-2)/10).

Mr. Reagan should be aware that he has not only provided us with a nifty sample problem for the programmer trainee, but also one which points out the need for every programmer to understand how computers represent numbers and the ways in which these representations can result in errors!

Keith S. Reid-Green, Director
Software Systems Development
Educational Testing Service
Princeton, NJ 08540

Seven points to consider before you buy your small computer.

In this magazine, alone, there are probably a dozen ads for small computers. New companies are breaking ground like spring flowers.

How, then, do you determine which computer offers the features you need most... at the price you can afford?

We'd like to propose seven basic questions to help you make an intelligent decision.

1. How complete is the computer system?

Many buyers of small computers are in for a rude awakening when they have to spend additional money for interfaces.

The Sol-20 Terminal Computer was the first *complete* small computer system. Everything you need to make it work is included in the basic package.

2. Is powerful system software available?

It won't do if your system is "tongue-tied."

Processor Technology Corporation has devoted more effort to the development of software than any other small computer maker. Our latest offering is the first fully implemented disk operating system for a small computer: PTDOS. It contains over 40 major commands, several languages and numerous utilities. Our high level languages include Extended BASIC, Assembler, FORTRAN,* FOCAL and PILOT.*

3. Is the system easy to expand?

More and more computer owners are expanding their small computers to handle business and other specialized requirements.

The largest Sol system can handle 64K bytes of RAM memory and operate with a three megabyte on-line disk memory. Sol systems use the S-100 Bus. So you can use a wide variety of hardware.

*Available soon.

4. Is the computer well-engineered?

Our Sol systems are the most conservatively rated and ruggedly built in the industry, period. In addition we designed them with you, the user, in mind; Sols are easy to build and a joy to operate.

5. Does it have proven reliability?

What is the track record? There are over 5,000 Sol systems in the field. Our track record for reliable performance is unparalleled in the small computer field.

6. Does it have good factory support?

A computer is a complex piece of hardware. So you want to be sure it is backed up with complete manuals, drawings and a factory support team that cares.

Processor Technology offers the most extensive documentation of any small computer manufacturer. And we maintain a patient, competent telephone staff to answer your questions.

7. Are maintenance and service people accessible?

Where are they located?

Processor Technology has maintenance and service people in over 50 cities around the U.S.

As you continue turning the pages, see how we stack up to the other computers in this magazine. If we've succeeded in whetting your appetite, see your Sol dealer or write for information on the complete family of Sol computers.

Processor Technology Corporation, Box C, 7100 Johnson Industrial Drive, Pleasanton, CA 94566. (415) 829-2600.

Processor Technology

put...input/output...in

Computer Club in Finland

Dear Editor:

I'm writing to you to make you aware of our Computer Club that was started some 3 months ago and to get it mentioned in the List of Amateur Computer Clubs.

We would be very happy to be able to contact clubs in the US and also to get information from manufacturers of equipment that could be hooked into the 6800 bus. We started our club by purchasing a MEK6800D2 Kit from our local friendly Motorola rep., were lucky enough to find a paper-tape reader and punch at a scrapyard. What we now need most is memory and a VDU terminal and would be very happy if somebody could tell us of suitable products, as it is not financially possible for us to get to the States to a Hobby Computer Fair and there are no Hobby Computer Shops around, either.

What we lack in availability of equipment we try to make up in enthusiasm and effort. Just think how many of you would have a Home Computer if you had to find out about them in foreign magazines written in a foreign language, too.

Kurt Soderstrom
Computer Club of Porvoo
Gammelbackantie 3E 33
SF-06100 Porvoo 10
Finland

CRTs and Their Effect on Users

Dear Editor:

In reference to the letter from Ken Roberts in your Sept-Oct 1977 issue, I doubt that he is suffering from CRT radiation, since he made no mention of such expected symptoms as hair falling out, nose twitching spastically, and eyeballs rolling uncontrollably. I suspect instead that Mr. Roberts is suffering from the subtle emanations given off by the S-100 memory bus ... a little-known malady, all research on which has been ruthlessly suppressed by a sinister combine of giant microcomputer manufacturers.

One alternative possibility is a case of *Sardonicus Counterculturus*, which is characterized by the tongue curling up in the cheek. Obviously the "Input-Output" staff has not yet contracted this disease ... beware, for it is known to be highly contagious.

I'm sending both you and Mr. Roberts copies of an article my partner Bonnie wrote concerning CRTs and their effects on users. This appeared some time ago in the CAPONE Newsletter (Chicago Area Programmers of NOVAs and ECLIPSE). I think it's relatively comprehensive; may give you a scrap or two of information.

Laurance F. Wygant
Chief Programmer
The Toolsmith Organization, Ltd.
Box 95094, Woodfield Mall
Schaumburg, IL 60195

Tips on Cathode Ray Tube TERMINALS

Bonnie Wygant (Pelam, Inc.)

The use of CRTs (cathode ray tubes) as a tool for on-line data entry is growing rapidly. They have come into use so fast that little is known about their effect upon the people who use them. The only reference I have seen to the subject is a recent article called "CRTs pose health problems for operators", by Dr. Olov Ostberg*. He mentions several areas in which complaints can be classified. The major difficulty is that the symptoms are very common (such as eyestrain, headache, burning eyes, body and mental fatigue) and are not taken as serious complaints.

Here are some tips to consider when working with CRTs. The key is to MINIMIZE FATIGUE! Ways to do this are:

1. Be sure the keyboard and screen are both at a comfortable height. A good height for the keyboard is very similar to that of a typewriter; hands should be able to rest on the keys about level or slightly downward from the elbow point. Your line of vision with the screen should be straight, since holding the head too high or too low can quickly make the neck muscles tired. An alternate possibility is to vary the level of your chair; however, your feet should rest flat on the floor and carry the weight of the legs, or you will cut off circulation. Good posture is essential. A proper chair is essential.

2. Try to have good light—too much can be as bad or worse than too little. The light near a window is generally too bright, as is the fluorescent lighting commonly found in offices. It produces a glare that is fatiguing. Fluorescent bulbs are notorious for flickering and this, too, is undesirable. Reflections on the screen are a no-no.

3. Clean the screen often with alcohol or glass cleaner, whatever the manufacturer recommends. Dust will make the image on the screen appear hazy, causing your eyes to constantly adjust to correct the focus. (This is an eye reflex of which we are never really aware.) This too causes fatigue. Cleaning at least once a week is a good practice.

4. Diversions from long periods in front of the CRT can be extremely beneficial. Break up your activities—chase a person of the opposite sex around the office at regular intervals. Make good use of break and lunch periods to remove yourself from the immediate environment. If you are a bookworm, remember that reading during a break may further tire your eyes.

5. Try to have your CRT out of the immediate vicinity of the computer. The noise of the inductors and the blowers is a mixture of sound that, combined with the heat generated from the machine, can make you extremely sleepy. A printer or card reader in use can be a very annoying distraction. Concentration becomes a struggle. The end result is fatigue.

6. For those who wear eye makeup it is advisable to try non-allergenic brands. Most have some allergies whether they are cognizant of them or not. A good test to find out if you are allergic is to put your eye makeup on... Do your eyes feel heavy?... Or like you have Plymouth Rock inside one or both eyes? Are your eyes red or extremely sensitive? Do you feel like rubbing them? You're probably slightly allergic. Non-allergenic cosmetics may cost more but believe me — they're worth it.

Let me say in closing that fatigue is a very general term, but the use of the word here covers both physical and mental reactions, as the two are often mixed. Try to develop good habits now.

* *Health and Safety*, Vol. 44 No. 6 (Nov/Dec 75) pp 24-26, 52, 50, 46.

U.S. POSTAL SERVICE

STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION

(Required by 39 U.S.C. 3685)

1. TITLE OF PUBLICATION: CREATIVE COMPUTING — A. PUBLICATION NO.: — 2. DATE OF FILING: 10/14/77

3. FREQUENCY OF ISSUE: Bi-monthly — A. NO. OF ISSUES PUBLISHED ANNUALLY: 6 — B. ANNUAL SUBSCRIPTION PRICE: $8.00

4. LOCATION OF KNOWN OFFICE OF PUBLICATION (Street, City, County, State and ZIP Code) (Not printers): 51 Dumont Place, Morristown, New Jersey 07960

5. LOCATION OF THE HEADQUARTERS OR GENERAL BUSINESS OFFICES OF THE PUBLISHERS (Not printers): 51 Dumont Place, Morristown, New Jersey 07960

6. NAMES AND COMPLETE ADDRESSES OF PUBLISHER, EDITOR, AND MANAGING EDITOR

PUBLISHER (Name and Address): David Ahl, 20 Lynnfield Drive, Morristown, New Jersey 07960

EDITOR (Name and Address): Steve Gray, 219 W. 81st St. Apt. 7C, New York, New York

MANAGING EDITOR (Name and Address): Burchenal Green, RFD 3, Highland Drive, Peekskill, New York 10566

7. OWNER (If owned by a corporation, its name and address must be stated and also immediately thereunder the names and addresses of stockholders owning or holding 1 percent or more of total amount of stock. If not owned by a corporation, the names and addresses of the individual owners must be given. If owned by a partnership or other unincorporated firm, its name and address, as well as that of each individual must be given.)

NAME	ADDRESS
David Ahl	20 Lynnfield Dr., Morristown, NJ 07960

8. KNOWN BONDHOLDERS, MORTGAGEES, AND OTHER SECURITY HOLDERS OWNING OR HOLDING 1 PERCENT OR MORE OF TOTAL AMOUNT OF BONDS, MORTGAGES OR OTHER SECURITIES (If there are none, so state)

NAME	ADDRESS
NONE	

9. FOR COMPLETION BY NONPROFIT ORGANIZATIONS AUTHORIZED TO MAIL AT SPECIAL RATES (Section 132.122, PSM)
The purpose, function, and nonprofit status of this organization and the exempt status for Federal income tax purposes (Check one)

☐ HAVE NOT CHANGED DURING PRECEDING 12 MONTHS ☐ HAVE CHANGED DURING PRECEDING 12 MONTHS (If changed, publisher must submit explanation of change with this statement.)

10. EXTENT AND NATURE OF CIRCULATION	AVERAGE NO. COPIES EACH ISSUE DURING PRECEDING 12 MONTHS	ACTUAL NO. COPIES OF SINGLE ISSUE PUBLISHED NEAREST TO FILING DATE
A. TOTAL NO. COPIES PRINTED (Net Press Run)	29,000	42,000
B. PAID CIRCULATION 1. SALES THROUGH DEALERS AND CARRIERS, STREET VENDORS AND COUNTER SALES	6,779	12,055
2. MAIL SUBSCRIPTIONS	18,086	20.406
C. TOTAL PAID CIRCULATION (Sum of 10B1 and 10B2)	24,865	32,461
D. FREE DISTRIBUTION BY MAIL, CARRIER OR OTHER MEANS SAMPLES, COMPLIMENTARY, AND OTHER FREE COPIES	3,000	3,240
E. TOTAL DISTRIBUTION (Sum of C and D)	27.865	35,701
F. COPIES NOT DISTRIBUTED 1. OFFICE USE, LEFT OVER, UNACCOUNTED, SPOILED AFTER PRINTING	1,135	6,299
2. RETURNS FROM NEWS AGENTS	0	0
G. TOTAL (Sum of E, F1 and 2—should equal net press run shown in A)	29.000	42,000

11. I certify that the statements made by me above are correct and complete. — SIGNATURE AND TITLE OF EDITOR, PUBLISHER, BUSINESS MANAGER, OR OWNER: Burchenal Green

12. FOR COMPLETION BY PUBLISHERS MAILING AT THE REGULAR RATES (Section 132.121, Postal Service Manual)

39 U. S. C. 3626 provides in pertinent part: "No person who would have been entitled to mail matter under former section 4359 of this title shall mail such matter at the rates provided under this subsection unless he files annually with the Postal Service a written request for permission to mail matter at such rates."

In accordance with the provisions of this statute, I hereby request permission to mail the publication named in Item 1 at the phased postage rates presently authorized by 39 U. S. C. 3626.

SIGNATURE AND TITLE OF EDITOR, PUBLISHER, BUSINESS MANAGER, OR OWNER

PS Form Mar. 1977 3526 (Page 1) (See instructions on reverse)

Look To The North Star HORIZON Computer.

HORIZON™— a complete, high-performance microprocessor system with integrated floppy disk memory. HORIZON is attractive, professionally engineered, and ideal for business, educational and personal applications.

To begin programming in extended BASIC, merely add a CRT or hard-copy terminal. HORIZON-1 includes a Z80A processor, 16K RAM, minifloppy™ disk and 12-slot S-100 motherboard with serial terminal interface — all standard equipment.

WHAT ABOUT PERFORMANCE?

The Z80A processor operates at 4MHZ — double the power of the 8080. And our 16K RAM board lets the Z80A *execute at full speed.* HORIZON can load or save a 10K byte disk program in less than 2 seconds. Each diskette can store 90K bytes.

AND SOFTWARE, TOO

HORIZON includes the North Star Disk Operating System and full extended BASIC on diskette ready at power-on. Our BASIC, now in widespread use, has everything desired in a BASIC, including sequential and random disk files, formatted output, a powerful line editor, strings, machine language CALL and more.

EXPAND YOUR HORIZON

Also available—Hardware floating point board (FPB); additional 16K memory boards with parity option. Add a second disk drive and you have HORIZON-2. Economical serial and parallel I/O ports may be installed on the motherboard. Many widely available S-100 bus peripheral boards can be added to HORIZON.

QUALITY AT THE RIGHT PRICE

HORIZON processor board, RAM, FPB and MICRO DISK SYSTEM can be bought separately for either Z80 or 8080 S-100 bus systems.

HORIZON-1 $1599 kit; $1899 assembled.
HORIZON-2 $1999 kit; $2349 assembled.

16K RAM—$399 kit; $459 assembled; Parity option $39 kit; $59 assembled. FPB $259 kit; $359 assembled. Z80 board $199 kit; $259 assembled. Prices subject to change. HORIZON offered in choice of wood or blue metal cover at no extra charge.

Write for free color catalogue or visit your local computer store.

NORTH STAR COMPUTERS
2465 Fourth Street • Berkeley, California 94710 • (415) 549-0858

COMPLEAT COMPUTER CATALOGUE

We welcome entries from readers for the "Compleat Computer Catalogue" on any item related, even distantly, to computers. Please include the name of the item, a brief evaluative description, price, and complete source data. If it is an item you obtained over one year ago, please check with the source to make sure it is still available at the quoted price.

Send contributions to "The Compleat Computer Catalogue," *Creative Computing,* P.O. Box 789-M, Morristown, NJ 07960.

BOOKS AND BOOKLETS

MICROCOMPUTER TROUBLESHOOTING MANUAL

The newly published *Microcomputer Troubleshooting Manual* is aimed at the owner of a microcomputer who is not an experienced technician, yet would like to do minor maintenance and troubleshooting on his microcomputer system. The manual assumes no experience or knowledge beyond that which the person who programs his system in machine language will already have. Basically, the manual tells you how to interpret the symptoms of an ailing system and presents a systematic technique for locating their causes. There is no theory here, just a few simple procedures to follow when things go wrong. Since most causes of system malfunction *are* relatively simple, this 20-page manual might be just what you're looking for. $5.

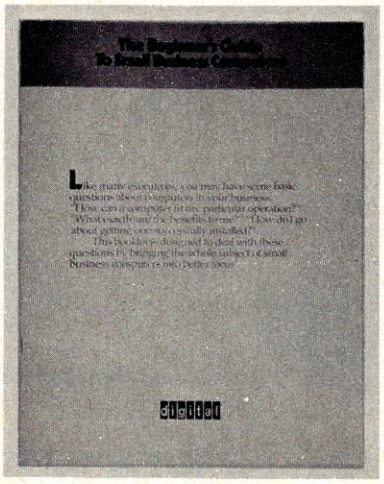
Pat Rankin. Micro-Info Assoc., Box 849, Castroville, CA 95012.

BRAIN TICKLERS

This 30-page booklet, "A Collection of Brain Ticklers for Everyone," is a collection of about four dozen problems (with solutions). A few are fairly well known among puzzlers, such as Crossed Ladders, but most are refreshingly new, and cleverly illustrated. $1.75.

Taylor Associates, 59 Middlesex Turnpike, Bedford MA 01730

VENDOR LITERATURE



BEGINNER'S GUIDE TO SMALL BUSINESS COMPUTERS

A new booklet from Digital Equipment Corporation, designed to unravel the mysteries of small business computing systems, details a step-by-step approach to matching computer capabilities with business needs.

Digital Equipment Corporation, Communications Services, Brochure EA 07430, 444 Whitney Street, Northboro, MA 01532.

HP MX SYSTEMS BOOKLET

A new 124-page booklet that includes nearly all published data sheets about HP 21 MX-based computer systems is available without cost from the Hewlett-Packard Company. The HP 21 MX family are high-performance small computers whose flexibility make them ideal as building blocks for scientific measurement and control systems, disc-based systems and as distributed network stations. Also discussed are memory systems, extenders, hardware and microprocessing accessories, and data communication, terminal and instrumentation interfaces.

Also available is a new 60-page catalog of all currently available real-time executive software for the 21MX computers and 1000 systems.

Inquiries Manager, Hewlett-Packard Co., 1501 Page Mill Road, Palo Alto, CA. 94304.

ORGANIZATIONS

MUMPS USERS' GROUP

The MUMPS Users' Group (MUG) offers literature and meetings on people interested on medical, educational, commercial, and other applications. In particular, a $2 "Book of MUMPS" is available, which contains a summary of the MUMPS language, information about implementations of MUMPS on the micro, mini, and large computers, 230 different MUMPS applications and their sources, names and addresses of over 40 MUMPS vendors, and document lists. Also available is the Standard MUMPS Pocket Guide at $1 per copy; this guide is a concise technical summary of the complete MUMPS language.

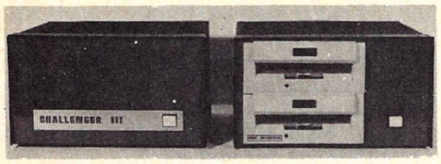
MUMPS Users' Group, 700 South Euclid Ave., St. Louis, MO 63110.

COMPUCOLOR USERS GROUP

Dedicated to the exchange of programs and technical data for the 8001, the Compucolor Users Group plans to issue a news bulletin, covering topics such as how to concatenate tapes and disks. For each accepted program, a member will receive in return a number of other programs. The initial membership fee of $10 covers duplicating and mailing costs. Among present programs are Blackjack, a "fantastically illustrated" version of Star Trek, illustrated slot machines, etc. The group hopes to exchange recorded media rather than program listings.

Compucolor Users Group, c/o S.P. Electronics, 5250 Van Nuys Blvd., Van Nuys, CA 91401. Send a large self-addressed envelope.

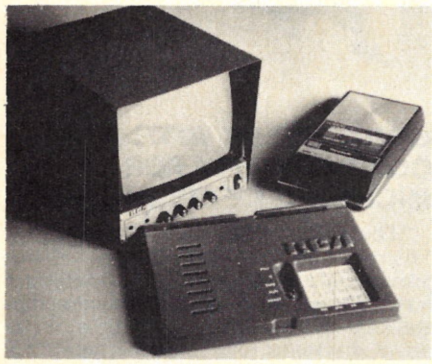
COMPUTERS



OHIO SCIENTIFIC CHALLENGER III

The Ohio Scientific Challenger III contains a new triple-processor CPU board "that can run virtually all published software available today for microprocessors at a very small cost increase over comparable single-processor computers." Equipped with three microprocessors, Challenger III runs 6800, 6502, 8080 and Z-80 programs. Challenger III comes standard with the OS-65D Disk Operating System. One application of the Challenger III is for personal-computing experiments with the three processors. Personal finances, strategic games, home and business applications are among the many applications. For advanced users a software processor status switch is available so that multiple-processor programs can be written. This option also includes a one-megabyte paper and user programmable vectors for both the 6502 and 6800 allowing real-time multitasking operation.

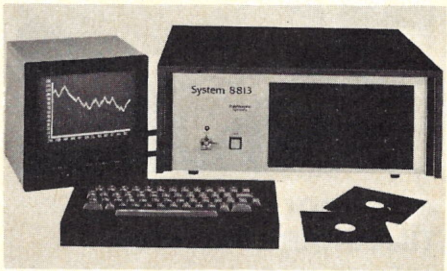
Ohio Scientific, 11681 Hayden, Hiram, OH 44234.



RCA HOBBY COMPUTER KIT

A new, expandable, low-cost hobbyist computer kit, called COSMAC VIP—Video Interface Processor— is now available from RCA Solid State Division to permit the hobbyist to assemble a microcomputer with which he or she can create and play video games, generate graphics and develop microprocessor control functions. The VIP is a complete computer on a printed circuit card, offering a complete operating system in only 4K bits of ROM. VIP's output directly interfaces with a monochrome CRT display or, when used with an FCC-approved modulator, a TV receiver. Programs can be generated and then stored in an audio cassette tape recorder for easy retrieval and use. The VIP features a single 8-1/2x11-inch PC card with the CDP1802 microprocessor, 2,048 byte RAM using 4K-bit static RAMs, single-chip graphic video display interface, built-in hexadecimal keyboard, 100-byte-per-second audio tape cassette interface and simple wall-plug regulated power supply. RAM capacity can be doubled on the card to 4,096 bytes by adding four 4K-bit devices, and can be expanded to a total of 32K bytes by adding further memory capacity through a 44-pin connector socket in the card. Parallel I/O expansion to 19 lines can be achieved on the PC card for such purposes as music synthesizers, relays, a low-cost printer or an ASCII keyboard. $275..

RCA Solid State Division, Box 3200, Somerville, NJ 08876.



POLYMORPHIC SYSTEM 8813

The PolyMorphic System 8813 is a compact complete disk-based microcomputer. The central unit, no larger than a stereo component, includes 16K bytes of RAM and room for three mini-floppy disk drives. Included in the package is a video monitor, keyboard with cable, and complete system software on diskette. System software allows you to put the system to work immediately, running applications in either assembly language or in fully extended BASIC. The small separate keyboard permits convenient use of the system at desk or table. The high-speed video display exhibits results in graphics and alphanumerics. Because it uses mini-floppy disks, the 8813 allows convenient storage and fast access to programs and data by means of simple user commands. Prices start at $3250.

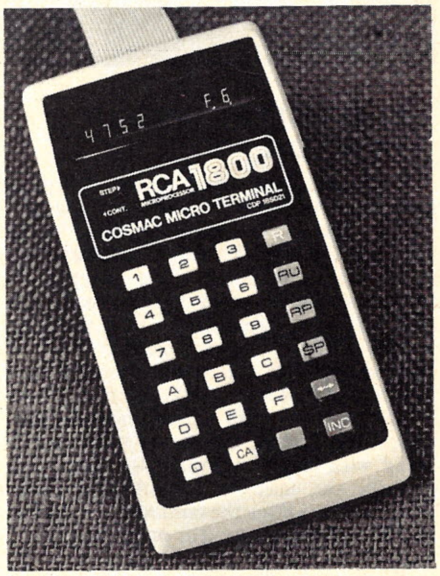
PolyMorphic Systems, 460 Ward Drive, Santa Barbara, CA 93111. (805) 967-0468

QUAY 8000 COMPUTER

The Quay 8000, a complete S-100 bus system based on the Quay 80 AI (Z-80 CPU) kit, includes a 12-slot mother board with all connectors and card guides, an 18-amp power supply, Quay 80 AI, Quay 80 SMB, 8K Static RAM board, wood grain cabinet, and 4K BASIC on cassette tape. The Quay 8000 is a complete kit requiring only a terminal. The popular Quay 80 AI CPU includes the serial I/O for RS232-C or 20-ma terminals, a jump-to-start monitor in UVPROM, sockets for up to 4K of UVPROM, 1K of static RAM, and a 2708 UVPROM programmer. $1095.

Quay Corp., Box 386, Freehold, NJ 07728. (201) 681-8700.

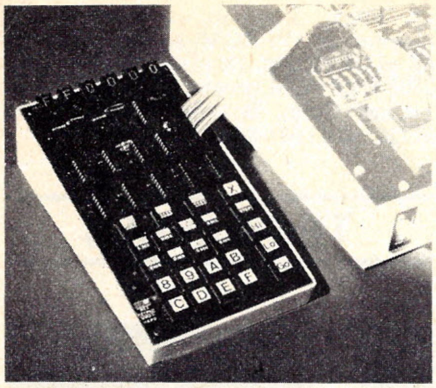
TERMINALS



HAND-HELD MICROTERMINAL

The size and shape of a pocket calculator, RCA Solid State's new COSMAC Microterminal CDP18S021 offers a low-cost, low-power, non-hard-copy alternative to the conventional, more expensive teletypewriter data terminal. Together with its Utility Program on an associated ROM, the Microterminal provides a convenient means of controlling a COSMAC microprocessor-based system, reading and modifying memory, and providing hexadecimal I/O capability. It is specifically suited for use with the COSMAC Evaluation Kit CDP18S021 or the COSMAC Development System CDP18S005, but it can readily be designed into user-built systems to provide control, communications, and debugging functions. Requiring less than 375 mA at 5V dc, the Microterminal is suited for portable or battery-operated applications. The Microterminal keyboard is used to enter data and memory addresses through the standard hexadecimal input keys. Eight control functions are regulated by seven additional keys and a switch which selects either continuous or single-cycle program execution. Specially designed for microprocessor applications, a new 600mA, 5V power supply, the CDP18S023, is also available. This supply is capable of powering both the Microterminal and the COSMAC Evaluation Kit augmented with four kilobytes of CDP1822S RAM. The power supply operates directly from any standard 115V 50/60 Hz outlet. The Microterminal is $140 andthe Microsupply $25.

RCA Solid State Division, Box 3200, Somerville, NJ 08876.



HEXADECIMAL KEYPAD FOR MMD-1

Two data entry and display keypads have been introduced by E&L Instruments, Inc. as accessories for their Mini-Micro Designer (MMD-1) training and development microcomputer. One keypad is furnished with a two-digit hexadecimal LED display expandable to six digits. MMD/HEX-1 and MMD/HEX-2 keypads provide the MMD-1 user with a low-cost, convenient method for programming with the 4-bit binary hexadecimal code. The simple calculator-type 16-key array with eight additional function keys permits the user to execute programs, modify or examine the contents of memory and registers and monitor program performance.

A factory-programmed HEX L/D PROM for hexadecimal conversion replaces the HEX PROM originally supplied with the MMD-1. One pair of 0.3-in. high LED hexadecimal displays is furnished with each MMD/HEX-2 keypad and two additional pairs are optional. Displays may also be added to the MMD/HEX-1 keypad.

The assembled MMD/HEX-1 is $125; $105 kit. The assembled MMD/HEX-2 is $185; $155 kit including the pair of displays.

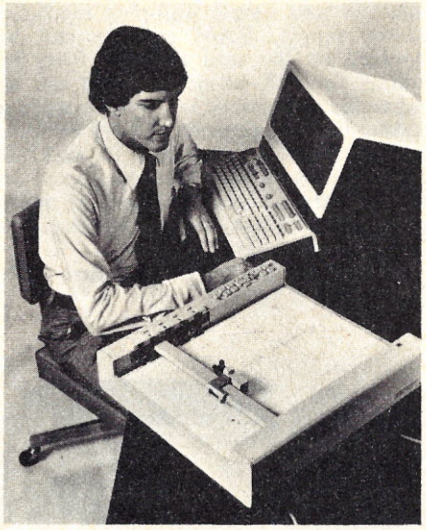
E&L Instruments, Inc., 61 First St., Derby CT 06418. (203) 735-8774.

TERMINAL FOR HOBBY MARKET

Anderson Jacobson, Inc. (AJ), has entered the hobby market with an I/O terminal priced for the serious hobbyist. Originally selling for over $4,000, the AJ 841 is an IBM Selectric terminal and off-line typewriter that has been completely refurbished. It includes a built-in ASCII interface. $995.

Steve Tritto, Anderson Jacobson, Inc., 521 Charcot Ave., San Jose, CA 95131. (408) 263-8530.

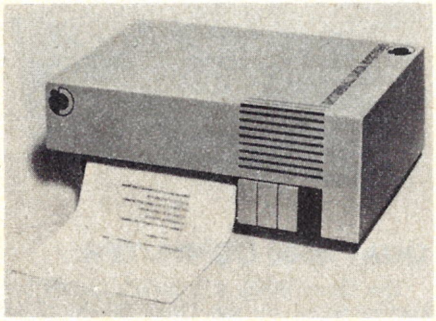
PERIPHERALS



MULTICOLOR GRAPHIC TIMESHARE PLOTTER

Microprocessorr-based circuits in this new ISO A3 (28 by 40 cm; 11 by 17 inch) flatbed timeshare plotter improve operating efficiency to reduce timeshare computer and transmission costs. The four-color Hewlett-Packard Model 7221A Graphic Plotter features internal arc and circle generation, and user-defined dashed line patterns. Any sequence of plotter instructions can be stored in the buffer memory as macro-instructions for recall as many times as needed. Up to 64 different macro-instructions can be stored at one time. Typical users of the Model 7221A include financial, administrative, scientific, engineering, and manufacturing personnel. Multicolor plots are especially useful in engineering modeling, plots of more than one measurement, manufacturing production control, numerical control verification' manpower loading, and in wide areas of mathematics, physics, and chemistry. Color graphics are achieved through programmable pen changing. $4600.

Inquiries Manager, Hewlett-Packard Co., 1507 Page Mill Road, Palo Alto, CA 94304.

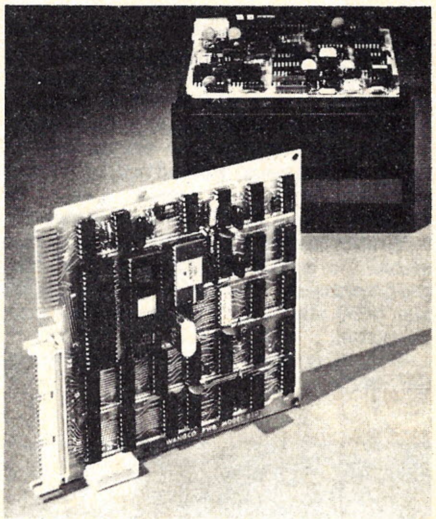


INTEGRAL IMPACT PRINTER

Integral Data Systems, offers a full-feature, dot-matrix impact printer designed for use with mini or micro-computer systems. Printing at rates to 120 cps with up to 132 characters per line, the Integral Impact's standard features include an RS-232 and current-loop serial interface, enhanced mode (double-width) characters and selectable character and line sizes. Multiple copy capability on both fan-fold and roll paper is also a standard capability. Serial baud rates of 110 to 1200 bits per second are selectable, and for those who prefer a parallel interface, that capability is also provided. Switch settings select character sizes and line length from 80 to 132 characters per line. $475..

Integral Data Systems, Inc., 5 Bridge Street, Watertown, MA 02172. (617) 926-1011.

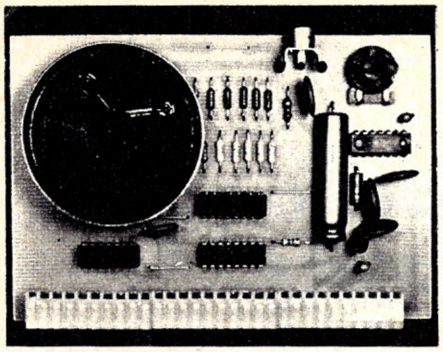
MISC. HARDWARE



CONTROLLER WITH S-100 COMPATIBILITY FOR MICRO-DISKETTE DRIVES

"The Wangco 8201 Micro-Controller for "micro-sized" 5¼-inch diskette drives provides a general-purpose host interface for 6800 and 8080-based microcomputer systems, minicomputers and other byte-oriented systems. One version of the 8201 is pin-compatible with the industry standard S-100 bus. A single printed-wire board plugs directly into the S-100 connector. The 8201 uses the Intel MCS 8048, a microprocessor providing 1K of ROM, RAM and I/O ports on a single chip. Formatting performed by the 8201 Controller is of a soft-sectored, modified IBM type of 16 sectors per track, 128 bytes per sector. The Micro-Controller provides for a 128-byte sector buffer or multiple sector transfer without buffering. Unique functions of the 8201 include a diagnostic command which causes self-test operations to be performed on all diskettes, and a duplicate function which automatically copies diskette information from one drive to another at one command from the host. The Wangco 8201 will control up to four drives. $490.

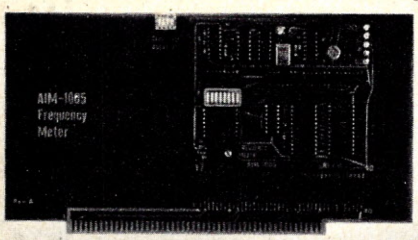
Wangco, Inc., 5404 Jandy Place, Los Angeles, CA 90066. (213) 390-8081.



MUSIC FOR SW OWNERS

The Newtech 68 music board enables the user to generate music, sound effects, rhythms, Morse code, and touch-tone synthesis. The 68 music board, designed for the Southwest Technical Products Corp. 6800 computer, comes fully assembled and tested. It consists of a digital-to-analog converter, audio amplifier, speaker, volume control and phono jack for convenient connection to an external speaker or home audio system. The Users Manual includes sound-effects programs, test routines, listings of a BASIC program for writing musical scores, and a 6800 assembly-language routine for playing them. An AC-30 compatible cassette contains programs from the Users Manual and software for pre-coded songs. $59.95.

Newtech Computer Systems, Inc., 131 Joralemon St., Brooklyn, NY 11201.



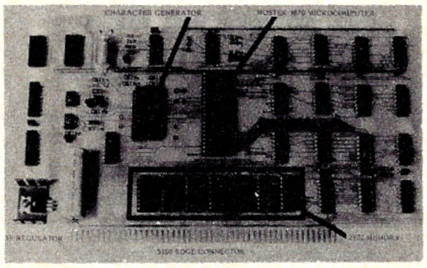
FREQUENCY METER

Automated Industrial Measurements (AIM) has announced a frequency meter, AIM-1005, that may be directly interfaced with 8-bit microcomputers. It has 13 bits of resolution plus overrange, and has eleven time-base ranges, from 10 microseconds to one hour. AIM-1005 is interfaced with the microcomputer as if it were memory, and can be located in 14 different locations. These locations are controlled with switches on the board, as is the time-base range. It is accurate to within ±1 count over the entire operating temperature range of 0°C to 70°C. The input frequency range is from DC to 25 MHz minimum. The input can be TTL, or the input comparator may be used to provide ±15 volts of common mode as well as a high-impedance input. An optional input is an AIM-1003, which allows one of AIM's transducer digitizers to be connected using only a twisted pair of wires. The input connections can be made from standoffs on the top of the board, or through the edge connector.

If real-time measurements are required, an external reset is provided which allows the start of the count period to begin when the reset is removed. The microcomputer is informed that a real-time measurement is taking place by having an external reset and status flags as part of the first data byte.

The complete frequency meter is on a 4" by 4½" PC card with a 40-pin edge connector with contacts on 0.1" centers. Because of its compact size, AIM-1005 is available mounted as a daughter board on the larger microcomputer cards, such as the Altair or IMSAI, offered as an option by AIM for $30 additional. AIM-1005 will also fit into any 4½" card-cage system, such as the Zilog Z80 or the Pro-Log systems. $178.

Barry Hilton, Automated Industrial Measurements Inc., P.O. Box 125, Wayland, MA 01778. (617) 653-8602.



VIDEO TERMINAL BOARD

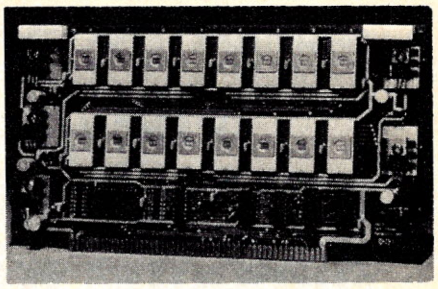
A low-cost video terminal board designed around the Mostek 3870 microcomputer from Vectron is sized to fit into S-100 systems, but may also be used in standalone applications. It features both ASCII and BAUDOT serial interfaces, and requires only the addition of a standard ASCII encoded keyboard and a TV monitor to configure a complete full duplex video terminal. Among the features provided are full X-Y cursor control (both absolute and relative), screen clear, clear to end of line, page mode and autoscroll, 96 displayable characters, 16 line by 64 character display, plus multiple baud rates up to 300 baud ASCII. In S-100 systems the SCT-100 may be powered directly from the unregulated+7 VDC bus (none of the other S-100 bus pins is used). For standalone applications an on-board rectifier and filter permit the operation of the board directly from an external 6.3 VAC, 1.0A transformer. Assembled and tested, $185. A partial kit including the PC board, the custom-programmed 3870 microcomputer, and the character generator ROM is $85, or a complete kit containing all necessary components is $155.

Vectron, Box 20887, Dallas, TX 75220. (214) 350-5291.

BIDIRECTIONAL I/O BOARD

Micrologic's M712 general-purpose bidirectional I/O board is specifically designed to allow the Phi Deck cassette system to be interfaced to an S-100 computer. The M712 requires only the cassette drive and control logic, both user-supplied, for operation with an Imsai, Sol or other S-100 computer. $99.95 kit; $119.95 assembled.

Byte Shops of Indianapolis, 5947 East 82nd St., Indianapolis, IN 46250. (317) 842-2983.



16K EPROM MEMORY KIT

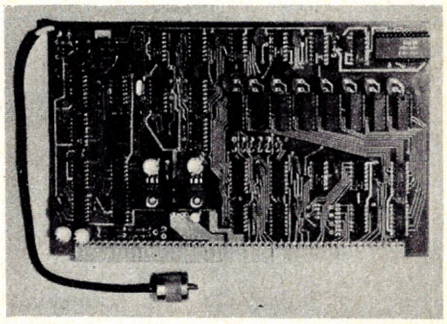
Designed to plug into the popular S-100 bus, the Ibex memory board holds 16,2708 EPROMs. Unused 4K sections can be disabled to allow RAM to exist within the board's address space. The board also has provisions for a wait state to allow it to run on a Z80 system. The kit comes complete with sockets for all IC's. $85, less EPROMS.

Ibex, 1010 Morse Ave., Suite 5, Sunnyvale, CA. 94086. (408) 739-3770.

IMPROVED M&R RF MODULATOR

The "SUP 'R' MOD" is an improved version of the RF modulator made by M & R Enterprises as described in the Sept-Oct issue of *Creative*. With the new unit, it is no longer necessary to modify your television set. The RF modulator is installed in your computer and an isolation switch mounted on the antenna terminals permits you to switch between normal usage of your television and the computer. The "SUP 'R' MOD" is recommended by APPLE Computer for use with its systems. "SUP 'R' MOD" is $24.95 plus $1.00 for postage and handling.

M & R Enterprises, P.O. Box 1011, Sunnyvale, CA 94088.

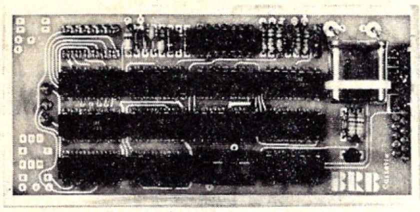


GIMIX VIDEO BOARD

The Gimix video board, assembled and tested for use in SWTP systems, generates 16 lines by 32 upper-case characters, and can be jumper-selected for 16x64 when used with a 10-MHz video monitor. A dual-port 1K RAM can be jumpered to the beginning of any 1K memory segment, which the processor can read or write as though the memory were part of the system. Text scrolling and cursor are generated by software. $249.

Gimix also has an 8K EPROM board for the SW 6800, and plans to offer a motherboard, relay-driver board, serial and parallel boards, etc.

Gimix, Inc., 1337 West 37th Place, Chicago, IL 60609. (312) 927-5510.



2400-BAUD CASSETTE INTERFACE

Microprocessor programs and data can be loaded and dumped from an audio cassette eight times faster than the standard 300 baud with the new Wince cassette interface. The interface also supports 300-baud Kansas City Standard operation. It interfaces directly to the Motorola 6850 ACIA. The 2½" x 5" module also contains an RS-232 interface for standard baud rates from 150 to 9600. $139.

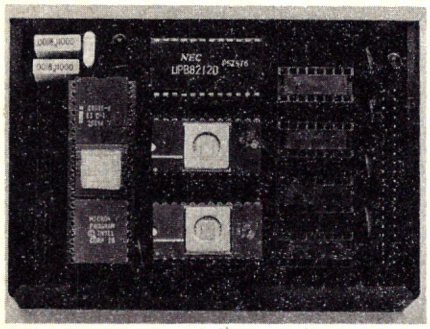
Wintek Corp., 902 N. 9th Street, Lafayette, IN 47904 (317) 742-6802.

ADAPTER CARD INTERFACES PERSCI DISK TO S-100

INFO 2000 has an adapter card that connects the PerSci Disk Drive Controller to any S-100 bus. The INFO 2000 Adapter now permits easy interfacing to the fastest disk drive available in the low-priced line. The INFO 2000 Adapter card includes circuitry for an additional 3K of EPROM and 1K of fast RAM. The advantage to this is that the user can place his entire operating system and scratch pad in the EPROM and RAM on the adapter and free up all his system RAM for programs. All that is required is that you purchase four 2708 EPROM and two 2114 RAM chips. The INFO 2000 Adapter kit can be assembled in one hour. A prototype area is included on the board for other features such as a wait state if you require one. INFO 2000 will provide all the necessary details for assembly and adding the wait state.

What the INFO 2000 Adapter owner can now have is a complete operating system (DOS), "intelligent controller," 7K of EPROM, 2K of RAM, all on one card that plugs into the S-100 bus. Kit, $85.

INFO 2000, P.O. Box 3196, Culver City, CA. 90230.



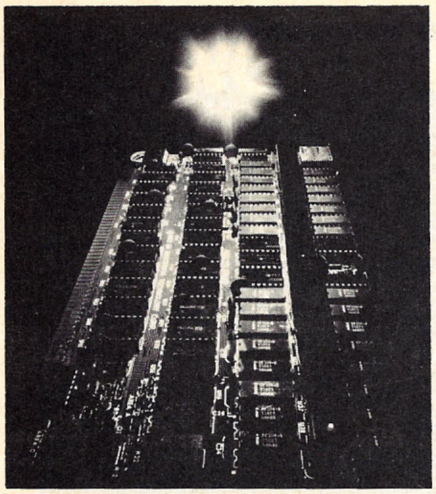
CONTROLLER IN A BOX

The CM-48 is a complete stand-alone computer controller. It contains its own crystal clock (for accurate time-keeping), 1K of EPROM for program storage, 64 bytes of RAM data storage, 27 I/O lines, a few wire-wrap sockets for relay drivers, etc., all in an aluminum case. The CM-48 is built around the new Intel 8048 microprocessor, fast (2.5us cycle time) and efficient (all instructions are 1 or 2 bytes long and require only 1 or 2 cycles). The processor includes a built-in 8-bit counter which can also be used as a precision timer.

The power, flexibility, and size of the Computer Module make it ideal for environmental control (such as heating and cooling based on time of day, inside vs. outside temperature, etc.). It also makes a great burglar alarm or remote security station. Or, it can be used as a universal interface box, such as RS-232 to parallel ports or as a flexible interface to DVMs and other test equipment. Computer Modules can also control robots, trains, automotive accessories, telephone dialers, slide projectors, etc.

The CM-48 with 8048, crystal oscillator, and 2708 1K EPROM, requires + 12v, ±5v; $149. The CM-48-1 with 8748 microprocessor and built-in 1K EPROM, requires 8-12 volts unregulated; $249. The CA-80/48 Cross Assembler is a full-feature 8K assembler that makes it easy to compose, edit, print and store programs for the 8048-based Computer Modules using any 8080-based computer system. $48.

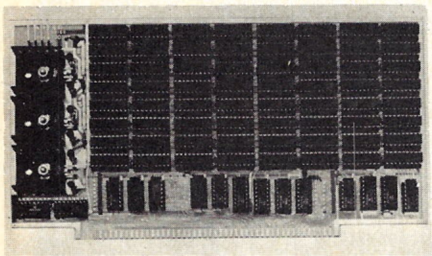
Oliver Audio Engineering, Inc., 7330 Laurel Canyon Blvd., North Hollywood, CA 91605. (213) 765-8080.



PROCESSOR TECH SEMIKITS

The new 16KRA RAM board from Processor Technology is the first in a series of memory modules offered in fully-assembled and wave-soldered form with pretested ICs. Semikits are priced about the same as competing kits according to the firm. Believed to be the first modules for small computers available in this form, the semikits memory modules will bring an end to such common kit-building problems as bad solder joints, heat-damaged components and faulty integrated circuits. Full documentation is provided with step-by-step procedures for the user to test and burn-in the boards. Features of the first semikit include a 16,384-byte memory, invisible refresh and worst-case access time of 400 nsec. Each 4096-word block is independently addressable for maximum system flexibility. Power is typically 5 watts, the same as most single 4K memory modules. Backup power connection is built-in. $369.

Processor Technology Corp., 6200 Hollis St. Emeryville, CA 98608. (415) 652-8080.



32K MEMORY BOARD

Artec Electronics has a 32K static memory board in modular form designed primarily for microcomputer hobbyists but also applicable for small business uses.

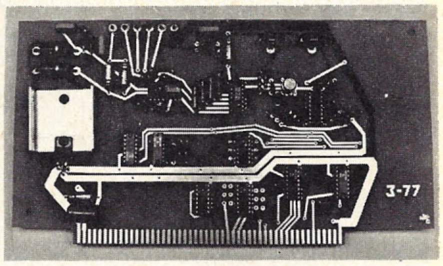
Designated 32K-100, the board is fully compatible with the S-100 data bus and speed-compatible with Zilog Z80-based systems. The basic board with all support circuits, power regulator, 8K bytes of memory and assembly manual sells for $290.

Additional 8K bytes of static RAM cost $255 each, and a full 32K memory board sells for $1,055.

The 32K-100 requires a single +8 voltage only, with a power usage of 3 amps; access time of the DMA-compatible board is 250 nanoseconds. DMA compatibility allows users to access memory directly without going through the central processing unit (CPU) on the microprocessor board.

The 32K board, fully buffered on all address and data lines, also features battery back-up (allowing operation at reduced power consumption), and a "bank select" provision so users can select the blocks, or banks, of memory they want to address.

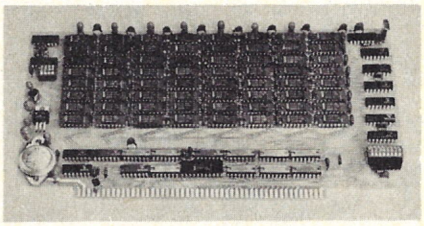
Artec, 605 Old Country Road, San Carlos, CA 94070. (415) 592-2740.



D/A CONVERTER

This self-refreshing four-channel D/A converter for the S-100 bus has instructions for alternate connection to Digital Group Systems, and includes complete output-port address decoding and tri-state bus connection. The converter is eight bits and can be set to cover any range within + 0 to 10 volts. The D/A converter operates at a rate of 50,000 conversions a second, updating the four sample hold outputs from a scratch-pad register. Circuit operation, as seen from the computer, is completely static—needing no software refresh or wait states — and is intended for use with output commands in BASIC programs. Computer checkout requires a simple four-byte program and calibration is limited to two minor adjustments. PC board, $34; kit with all components, $69; completed and tested board, $99.

Pinnacle Products, Box 3155, Talcottville, CT 06066



64K RAM BOARD

A fully-tested 64K RAM board offered by Extensys meets S-100 computer interface specifications, including Altair and Imsai units, and allows memory addition up to 1,048,576 bytes. The 5"x10" PC board contains 65,536 bytes and has hardware provision for bank-switching to add over one million bytes, far beyond normal microprocessor capacity. The board also allows memory address to be set in 8K-byte increments and provides hardware-write protection in 16K-byte increments. Voltages are +12 at 300 mA, +5 at 750 mA, and -5 at 1 mA. Cycle time is 500 nsec, with 400 nsec access time. Memory overlap protection is provided to ensure no conflict with existing memories. All boards are fully assembled, tested, and burned in. The 64K RAM board is $1,495. A 32K board is offered at $895, and a 48K board at $1,195.

Extensys Corp., 592 Weddell Drive, Suite 3, Sunnyvale, CA 94086. (408) 734-1525.

SOFTWARE

WORD-PROCESSING SYSTEM

Interactive Data Systems has developed a word-processing system, IDSWORD1, designed to run under MITS Disk Extended BASIC. Some of the more important features of the system are line editing, global editing, merging, reformatting, moving text, printing and form letters. IDSWORD1 is a package consisting of several programs. This fact is transparent to the user but allows it to run on a computer with 28K of memory. The user selects the mode of operation from a menu list and the control program executes the appropriate program and gets control back to select another mode. The system is provided on a diskette. $250.

Interactive Data Systems, Box 290, Owings Mills, MD 21117. (301) 486-6945.

"XIM" FOR KIM

"XIM" (Extended I/O Monitor) by Pyramid Data Systems is a programming-debugging package developed for the KIM-1 Microcomputer. XIM features a repertoire of 17 commands, four of which are user-defined for expansion. XIM can do operations such as block search, block move, block compare, set break points, display processor registers, and more. XIM resides in less than 1K of core at 2000 to 23FO(hex) and it is ROM-able and easily relocated to suit the user.

The cassette and paper-tape versions are $12 and $10 respectively. A 45-page manual with commented listings is included, shipped postpaid USA. NJ residents add 5% for sales tax.

For more information, send an SASE to: Pyramid Data Systems, 6 Terrace Ave., New Egypt, NJ 08533.

ANSI STANDARD FORTRAN IV

Technical Design Labs announce what is said to be the first complete ANSI STANDARD FORTRAN IV for a microcomputer." FORTRAN IV was written exclusively for Technical Design Labs and the Z80 by Small Systems Services, Inc. FORTRAN IV is equipped with many extensions that exceed the usual ANSI standard requirements: data types (integer*1, integer*2, real, doubleprecision, complex, logical and string), named common (equivalence, statement functions, implicit), full type conversion, full library of scientific and string functions, linking loader with automatic library search (can link with assembler output), full formatted I/O, sequential and direct-access I/O, hex constants, and control over placement of data and code areas (the code can run from ROM). This FORTRAN is a disk-oriented system. It runs in less than 24K with DOS, and both FDOS IV AND CP/M versions are available. The package now includes both the floppy diskette with

object code and a user's manual. $349.

Donna Galletti, Director of Sales, Technical Design Labs, Inc., Research Park, Building H, 1101 State Road, Princeton, NJ 08540. (609) 921-0321.

NORTH STAR EXECUTIVE SOFTWARE

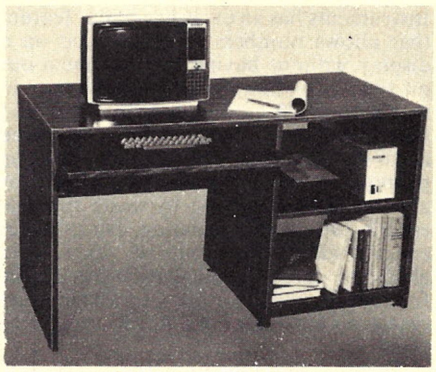
XEK, a complete system executive package for North Star users, available from the Byte Shop of Westminster, contains a disassembler capable of creating files that may be left in memory when changing from the disassembler to the executive package for re-assembly. The monitor software can accept input from cassette tapes and paper tape as either source or object files, as well as from the North Star diskette system. In addition, the assembler features a new auto-line editor for the creation of source files. This editor also extends to the modification of existing object files. Another feature is the XEK's ability to handle up to six named files at once that may be consecutively assembled to form one object file. $48.

The Byte Shop of Westminster, 14300 Beach Blvd., Westminister, CA 92683. (714) 894-9131.

MISCELLANEOUS

MICROCOMPUTER DESK

A desk specifically designed to house a personal or business microcomputer



system is now available from Computer Systems Design. The Microdesk can be assembled without tools in five minutes. The desk is constructed of high-density vinyl clad board and measures 48″ by 24″ by 28″. A sliding shelf at convenient typing height is provided for a keyboard. Also included for equipment and books are two fixed shelves, one of which is adjustable. $96.50, FOB Wichita.

Computer Systems Design, 1611 E. Central, Wichita, KS 67214.

MICRO-COMPUTER REPAIR SERVICE

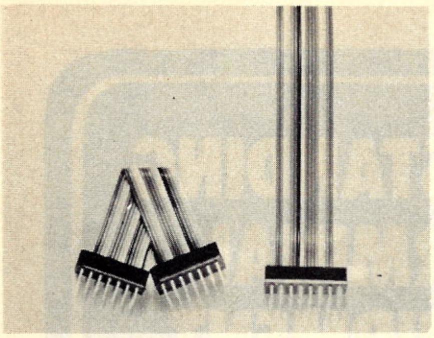
A new repair service called The Computer Doctor, Inc., has opened for business. The company offers a full range of repair and counseling services on all brands of microcomputers, processors, peripherals, and memory, and promises two-day service in most cases. As an added feature The Computer Doctor has a unique rebuild-it-yourself plan for kit builders who are having trouble. The Computer Doctor is Grant Snellen, a computer engineer with over 15 years experience in the industry, most of it in field services. He has worked for Harris and Digital Equipment Corporations and was chief engineer in charge of the design and construction of M.I.T.'s Wallace Observatory. The Computer Doctor is open 6 days a week, from noon to 10 PM.

The Computer Doctor, 12 Howard, St., Cambridge, MA 02139. (617) 661-8792.

S-100 CARDFRAME PANEL SET

Objective Design announces the availability of a starter set for the construction of an S-100 cardframe and power supply. The set includes front and rear panels of 1/8 inch aluminum, prepunched and drilled for maximum design versatility. Complete instructions for the assembly of both a full-sized 22-slot cardframe and a ten-inch 12-slot model are included. Also included is information on all required parts and sources of supply, full assembly instructions, and detailed plans, specifications, and illustrations. $39.95. All panel sets will be shipped insured freight collect FOB Tallahassee, Florida.

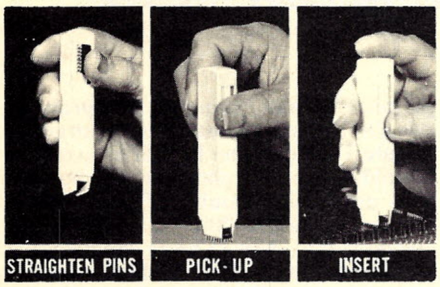
Objective Design, Inc., Box 20325, Tallahassee, FL 32304.



STOCK CABLE ASSEMBLIES

Interconnect Cable Assemblies are perfect for testing or connecting circuitry within a board or for jumping from board to board. Assemblies feature rainbow-color-coded flat cable with 26 AWG stranded conductors soldered and epoxy encapsulated to popular top-entry plugs. Plugs are available in 14 and 16-pin DIP configurations to fit all standard sockets and hole patterns. Pins are gold-plated phosphor bronze for performance and durability. Double-ended configuration available in 2", 4", and 8" lengths. Single-ended assemblies are offered in 12" and 24" sizes. Prices from $3.75 - 4.35.

In stock at your local electronics distributor or direct from O.K. Machine and Tool Corp., 3455 Conner St., Bronx, New York 10475. (212) 994-6600.



DIP/IC INSERTION TOOL WITH PIN STRAIGHTENER

The OK INS-1416 DIP Insertion Tool inserts both 14 and 16-pin IC packages into sockets or predrilled boards, with one-hand operation. Narrow profile permits tool to work on densely-spaced patterns, while insertion mechanism assures accuracy as well as excellent "feel." The tool includes a pin straightener built into the handle. Insert the IC, rock it on the straightening saddle, and push down on the tool. An automatic ejector delivers the IC ready to be placed in the insertion end for installation in your board or socket. $3.49.

O,K. Machine and Tool Corp., 3455 Conner Street, Bronx, NY 10475.

STAR TREK

STAR TREK IN COBOL

Rick Jones has a COBOL version of Star Trek, to be run under IBM CICS on a 3270 CRT, using special CICS maps "which play very well." It is based on Bob Leedom's Super Star Trek, with modifications and enhancements to exploit the advantages of a CRT, and is composed of "top-down" modules; it can be easily modified for individual tastes. The COBOL source code and CICS maps amount to about 3000 lines. $5 for full source listings; $20 for 9-track magnetic tape ($5 refund if you provide tape, or when you return Rick's tape).

Rick Jones, 15702 Horace St., Granada Hills, CA 91345.

CALCULATORS

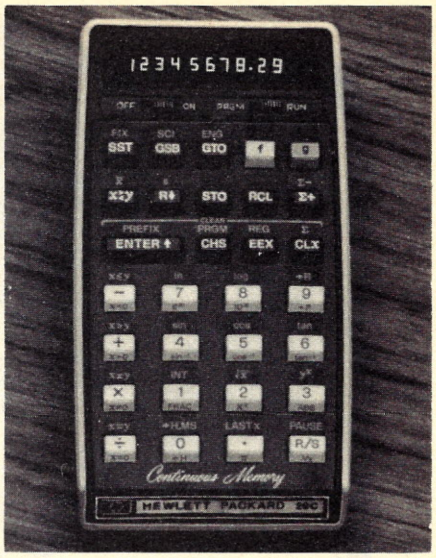
REPLAY PORTABLE CALCULATOR

A new portable calculator from Texas Instruments has an exclusive replay feature that allows numbers to be recalled on a display without having to print them out on tape. The TI-1680 permits the user to recall up to 20 entries, both number and function, from a memory stack to check accuracy and for editing and updating purposes. Each entry can be changed or altered without having to waste time re-entering an entire calculation. The user sees exactly what's been entered; what effect changing part of a calculation has on the final result, and he is able to recall incorrect entries to make changes. $29.95, suggested retail.

Texas Instruments Inc., Inquiry Answering Service, Box 53 (Attn: TI-1680), Lubbock, TX 79408.

HP CALCULATOR WITH CONTINUOUS MEMORY

The new Hewlett Packard HP-29C calculator features 98 fully merged program steps, full editing functions and continuous memory. Designed for use by engineers, scientists, technicians, surveyors and students, the HP-29C offers, at a keystroke programmable price, many of the features usually found only in fully programmable calculators, including three subrouting levels, 10 addressable labels and indirect addressing. Fully merged programming allows as many as four keystrokes to be combined into a single step of memory. CMOS memory chips allow programs and data be retained in



memory for long periods after the calculator has been turned off, with little battery use. In addition to its programming and editing functions, the HP-29C has all of the standard functions of HP advanced scientific calculators. $195.

A handheld printing version of the HP-29C, the HP-19C, will be available at $345.

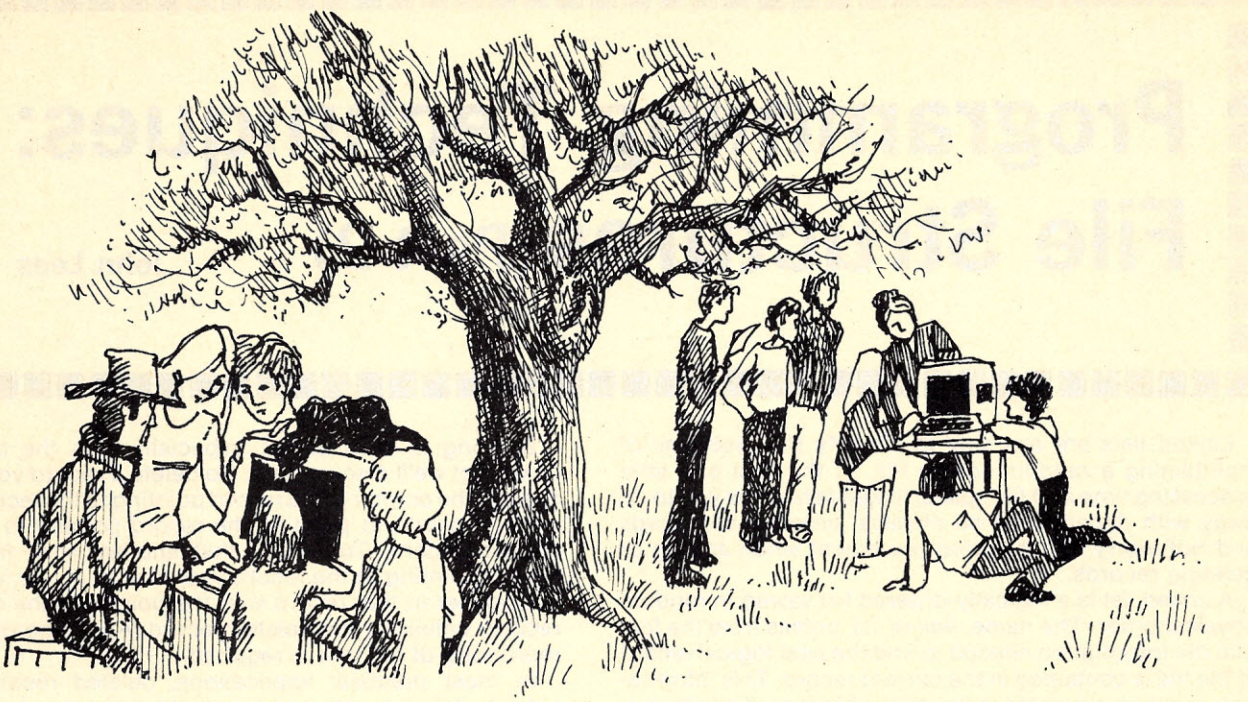
Inquiries Manager, Hewlett-Packard Co., 1507 Page Mill Road, Palo Alto, CA 94304.

HP CALCULATOR/WATCH

A new instrument that resembles a digital watch, is smarter than many pocket calculators, and performs more than three dozen functions, was introduced by Hewlett-Packard. The six-ounce HP-01 has six interactive functions: time, alarm, timer/stopwatch, date/calendar, calculator and memory. Twenty-eight keys (six finger-operated, 22 stylus-operated) enables the user to operate the HP-01. It also features 12 different display modes or indicators. The information is displayed with bright light-emitting-diodes. Price of the HP-01 is $650 for the stainless-steel model and $750 with gold-filled case.

Hewlett-Packard Co., Inquiries Manager, 1501 Page Mill Road, Palo Alto CA 94304.

Daring ideas are like chessmen moved forward; they may be beaten, but they may start a winning game—Goethe



Since 1972 we've been a place where people and computers get together.

Do you know anyone who wants to learn how computers work? Or how to program? Without masses of technical jargon? Then send them to *People's Computers*! We're a bimonthly magazine devoted to demystifying computers. Our pages aren't padded with ads–they're packed with articles, tutorials, interviews, listings, games, reviews and letters. *People's Computers* covers:

Computerese - basic principles about how computers work; tutorials on how to program.

Education - ready-to-use programs; tips on how to write computer-assisted educational materials; reports on educational uses of computers.

Blue Sky - Tiny BASIC and Tiny PILOT were born in our pages; there's now talk of robots and . . .

People & Computers - examination of the growing impact of computers on our lives and society.

Fun & Fantasy - computer games; whimsical odds and ends; and our swashbuckling hero–FORTRAN Man!

Some Comments about *People's Computers:*

"One of the first, if not *the* first, of the people-oriented computer publications . . . At $8 a year, it is a good bargain. You'll like it." *Kilobaud*

"The editorial flavor is . . . intended to be readable and enjoyable for the neophyte. Our resident non-computer people at *Byte* grabbed the first issue so quickly that it became difficult to find a copy . . . Should be sampled to be believed." *Byte*

"*People's Computers* has been plugging away for years as a force for good in the community". *Personal Computing*

"For the novice as well as the experienced computer user . . . an informal style, many useful annotations . . . a sounding board for novel ideas." *Computer Notes*

"Here's a first rate magazine for people who want to know about minicomputers and microcomputers . . . Powerful, low-cost technology is taking off – keep up with it by reading *People's Computers*". *The Workbook*

people's computers

Please start my one-year subscription (six issues) to *People's Computers* and bill me for just $8.

NAME ______________________

ADDRESS ______________________

CITY/STATE ____________ ZIP ____________

Outside the U.S., add $4 for surface postage. Airmail rates on request.

Unconditional Guarantee: If you ever wish to discontinue your subscription *for any reason*, we will refund the complete amount for the remainder of your subscription.

☐ Visa Card Number ______________

☐ Mastercharge Expiration Date ______________

Send this form or a facsimile to: *People's Computers,* Dept 54, 1263 El Camino Real, Box E, Menlo Park CA 94025

Programming Techniques: File Structures (Part 2)

John Lees

Linked lists are one way to simplify the problems of maintaining a random-access file. At the cost of a little processing time and file space, linked lists allow you to do away with dependence on physical ordering of records and not worry about record placement when adding or deleting records.

A linked list is a logically-ordered list stored in random physical order. The name, *linked list,* comes from the fact that the information needed to find the next logical record in the file is contained in the current record. This information is called a *pointer.* A linked list file record might look like this:

POINTER	AUTHOR	TITLE

and a file like this:

2		
0	HEINLEIN	DATA
3	ASIMOV	DATA
1	GUNN	DATA

The special first record simply contains the pointer (block number, sector number, or memory address in a practical implementation) to the first real record in the file. After getting that information, a program can access the file in logical order by following the pointers from record to record. By convention a "0" is used in the last record to indicate that it points to no other record. Our sample file is in alphabetical-by-author order and the pointer is a record number.

Now this might seem to be rather cumbersome until you consider the process of adding or deleting a record. To add, for instance, BESTER to our file, we simply stick the new record on the end of the file or in a hole left by a deleted record, determine that the new record goes logically after ASIMOV, take the pointer from the ASIMOV record and put it in the BESTER record, point the ASIMOV record at the BESTER record, and we're done. No fuss, no muss, no ugly resorting and moving records around in the file. The file now looks like this:

2		
0	HEINLEIN	DATA
4	ASIMOV	DATA
1	GUNN	DATA
3	BESTER	DATA

Deleting is even easier, especially with the double-linked list we'll discuss next. To delete a record you must change the pointer in the record pointing to the record you want to delete to point to the record pointed to by the deleted record. To phrase that another way, find the record pointing to the record you want to delete, replace the pointer in that record with the pointer in the deleted record, unlinking the deleted record. Again, no physical reordering of records is required.

In most practical applications, deleted records are actually linked together to form a "file" of deleted records. Then when you want to add a record, you find space for it by taking a "record," now just a chunk of space, from the file of deleted records and using it for the new record, deleting it from the file of deleted records and adding it to the real file. If this method is used, a newly-created file is initialized so that all the file space is records on the deleted records file. Of course if records are not all of the same length, that won't work, and things get considerably more complicated, but we won't go into that here.

Nothing says that a linked list record can have only one pointer and there are considerable advantages to devoting space to at least two pointers. A *double-linked list* has pointers in both directions so that the file can be searched either forward or backward. Being able to find both preceding and following record from the current record also greatly simplifies deletion. Double linking our sample file gives:

2	1		
0	3	HEINLEIN	DATA
4	0	ASIMOV	DATA
1	4	GUNN	DATA
3	2	BESTER	DATA

The second set of pointers point to the previous record and we've added to the special record a pointer to the last record in the file so that the file can be searched backwards. Adding a record to a double-linked list is much the same as to a single-linked list; you just have two sets of pointers to juggle.

A double-linked list can be made a *ring* by pointing the last record at the first record and the first record at the last record. That way processing can literally go around in circles without ever having to start again at the beginning.

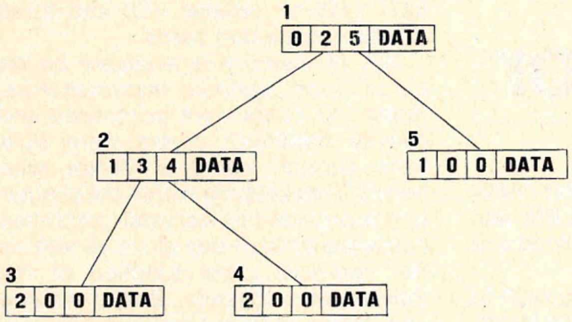
A common use of the linked-list organization is to represent a tree structure in a file. Trees find great computer-science application in game stategy and in some types of data bases. With three pointers in a record, you can define a binary tree (a tree with two branches at each node) that can be traversed in either direction. An n-ary tree can be defined by adding more pointers.

To illustrate a simple binary tree file:

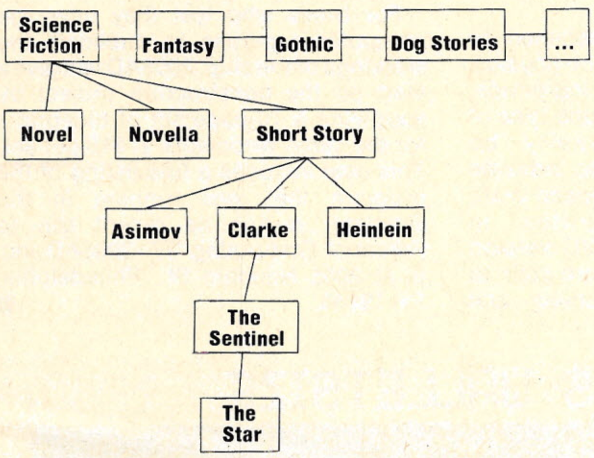
Binary Tree Record

PREVIOUS NODE	LEFT NODE	RIGHT NODE	DATA

Tree Structured File



To give some hint as to the possibilities of a data base built around the structure of a tree (usually called a hierarchical data base) consider the following diagram:



To stretch your mind even further, consider that it is possible to keep a linked list in more than one logical order. This is possible because the physical order of the records is totally irrelevant. You can, by having a pointer associated with each key field, have a file of names and addresses linked in both alphabetical and zip-code order, for instance. This can be quite handy in some applications and is impossible with a physical sequential file.

It is also possible, andquite usual, for a record in a linked-list data base to be on more than one list at a time. In the above example of a hierarchical data base, it is certain that some authors, and some books, would appear under more than one primary index level (Fantasy, Gothic, etc.). Why duplicate the records? Just attach the same record to different lists existing under different categories. This can become very complicated. The design of such data bases is not a task to be undertaken lightly.

To sum up linked list files: they are best suited to applications that require very little straight sequential processing and a lot of add or delete processing. Linked lists also lend themselves well to representing non-linear file structures and are heavily used in data-base applications. ■

SFS Walletsize Space-Shuttle Simulator

The SFS Walletsize, shown for the first time at Personal Computing '77 in Atlantic City, is a space-shuttle simulator, designed and constructed by a Philadelphia company with the appropriate name of 2005AD, Inc. The cabin itself, made of wood and metal, stands only five feet tall and is about four feet square, but the illusion is one of the grandest yet seen at personal computer trade shows.

As you sit down in the Pilot's seat, you roll yourself forward and become totally involved in the dynamics of the

At computer shows, the staff of 2005AD wears futuristic spacepeople costumes.

cabin. Overhead, lighting panels from the Apollo program wink on and off, while to your left and right aircraft instruments display more information than you can mentally process. The major focus of the display is the three TV screens (one a 25-inch high resolution monitor) that display three different types of computer-generated video.

The Walletsize is flown using a joystick, three slide controls and numerous switches, and the name of the game is simulated space flight. For the novice, one program will allow you to try landing your craft on a lunar platform. The center screen displays horizontal and vertical motion, the left screen displays a side view of the platform and the right screen displays an overhead view of the platform. This is probably the first time that a three-dimensional simulation of lunar lander has been played using a microprocessor. Miss the landing platform and the screen displays caustic comments on the depth of the crater produced, the cost of the LEM just destroyed and the amount of blood and gore produced.

If flying it yourself is not enough to satisfy your appetite for space travel, the 2005AD personnel will demonstrate a more complex three-dimensional docking program, involving not only roll, pitch and yaw, but also X, Y and Z coordinates. Approach is controlled by thrusters and retro engines, and the entire simulation is scored on accuracy, timing and fuel consumption.

The Walletsize is programmed to reflect the realities of flight physics, and as such is a great learning device. The craft is not limited to one type of simulation, it can easily be reprogrammed in BASIC to simulate various types of air and space craft. The simulation is written in PolyMorphic Systems BASIC Version A00. Within this program are calls to machine-language subroutines and PEEK and POKE statements to control the interrupt structure and I/O devices.

Peripheral chips and cards on the bus include a UART (TTY I/O), four ACIA (terminal and printer I/O), four CTC programmable counter/timers, one USART (tape I/O) one PIA (16-bit parallel I/O with handshake), sixteen 8212 (128-bit parallel I/O) and three PolyMorphic video cards.

40K of memory is available on 4K cards from various manufacturers. Some I/O cards were purchased and heavily modified; others were built from scratch using a Vector wire pencil. Timekeeping within the simulation is provided by a crystal-controlled hardware time-of-day clock as well as the real-time clock function of the PolyMorphic System Basic. Analog and audio output functions are performed by the Zilog CTC chips and by interface circuits designed by 2005AD. Switches and lights are interfaced through parallel ports and custom-designed circuits. All cards are connected to the S-100 bus in a modified IMSAI chassis. The CPU is a modified Z-80 card running at 2 MHz.

For those who feel they have the computer expertise to duplicate the workings of the ship, 2005AD will give a start on the construction project by supplying a 10-page set of blueprints for the cabin woodwork at $15 per set. This cost includes a free listing of the program that was running in the simulator during its recent trip to Personal Computing '77. 2005AD Inc. is at 2005 Naudain St., Philadelphia, PA 19146. ■

Many of the controls and lights in the SFS Walletsize are for effect only, but the three video screens are for real; they display Lunar Lander in three-dimensional simulation.

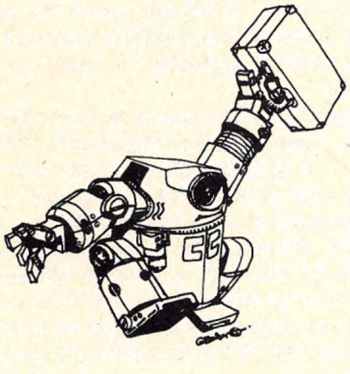
Profile of Three Smart Electronic Games...

In all three games, it's them against you, they know where you are, and they're out to get you.

Auto Race

Missile Attack

Football



Stephen B. Gray

With Milton Bradley's Comp IV (reviewed in the previous issue) you can take your time guessing the random number generated by the microcomputer inside the case. But with these three games from Mattel, they — the cars, the missiles, or the opposing team — are coming at you at full tilt, and you'd better pay full attention, right here and now. There's no time to fool around; you've got to act immediately. So if you're easily frustrated, or unable to deal with fast-moving real-time games, stay away from this trio, because these three little hand-held games soon separate the men from the boys, and the boys from the children.

Description

All three of these Mattel games are about the size of a hand-held calculator, 1x3x5 inches, weigh only a few ounces each, and use a nine-volt battery; each has a jack for an AC adapter.

Auto Race and Missile Attack are similar in that both use the Rockwell B6000 calculator chip, with some of the input/output lines modified. Both use a vertical "LED stick" display with three rows of seven vertical segments each, topped by a two-digit segmented numerical display. A three-position slide switch at bottom right determines in which lane you drive (Steering Control), or from which of three sites you launch your anti-missile missiles (Guidance Control). Auto Race has a Start/Reset switch; in Missile Attack it's called Arm/Off. The third control is quite different: in Auto Race it's a slide switch with four Gear positions, 1 through 4, which determine how fast the cars move; in Missile Attack, it's the Fire button, which launches your rockets skyward.

Football has more complex rules and a much more sophisticated display, so it requires the Rockwell B6100 chip, which has more ROM and more I/O. The playing field is a horizontal LED stick, with three rows of nine horizontal segments each. Pressing the ST key lights the Status display, showing down (1 through 4), field position (where the ball is, plus a symbol that shows which side of the 50-yard line the ball is on), and yards to go. Pressing the SC or Score key shows the home-team score, timing remaining in the quarter, and the visitor score. Either the ST or SC key resets the field for the next play, which involves a running back making his way down the field against five defensive tacklers. The up-arrow and down-arrow keys move the running back to his own left or right; the left-and-right arrows moves the running back toward the opposite goal, the direction depending on whether he's on the home team or visitor team. The K or Kick key may be used only on 4th down, and will kick between 1 and 65 yards; the computer decides whether to punt or to make a field goal. The last control is a three-position switch, for off, PRO 1 or PRO 2. In PRO 2 mode, the defense is programmed to react 50 percent faster than PRO 1.

Missile Attack

Basic Operation

Auto Race is simple, just "race your car up the track avoiding collisions with oncoming cars." (Sounds more like a Demolition Derby than an auto race.) You have to set the gear switch in first before the game will start. Your car is the bright blip that starts at the bottom of the track and moves up. The "collision cars" (dimmer blips) start at the top and move toward you, one or two at a time, but never three at a time, fortunately. You move the steering control left or right to avoid a collision. If one of the collision cars hits you, it pushes you back toward the starting time, and you lose time on the "Lap-Time Readout." After you finish a lap, your car is automatically returned to the starting line, ready for the next lap. In higher gear, the cars move faster, and you've got to react faster. At the end of four laps, the timer stops, you read out your time, and compare it with your previous best time, or with your opponent's.

Missile Attack is played almost the same way, with several important differences. The enemy missiles come in, from the top, at only one speed, and they can change direction. If only one of the 20 attacking missiles hits your city, at bottom center of the display, you lose and the game's over. You can protect your city by always launching your anti-missile missiles from the center position, but this will give you a minimum score. The game's object is

"to intercept as many of the 20 attacking missiles per game as high in the sky as possible." If you hit an attacker in the 6-mile zone (top of the screen), you score 6 points; a hit in the 5-mile zone and get 5 points, etc.

Football is meant for two players, although a child too young to understand the rules of the game could play it all by himself on the basis that he must shift back and forth, waiting for an open path down the field, and, because the display field is only 9 yards long, if he has to go 45 yards for a touchdown, he must run the whole length of the field five times to score. If he makes it to the end of the field in a single run, the computer automatically returns him to the start so that he may continue running. However, the opposing players can move from side to side as well as forward or backward, and as the running back scoots down the sidelines, he may be tackled from the front, the side, or from behind. The player who made the tackle will blink on and off, to show his position. Automatic scoring provides 7 points for a touchdown, 3 for a field goal. Just as in real football, by the way, you get four downs to go 10 yards.

Sound Effects

All three of these games contain what the schematics refer to as a "piezo-ceramic speaker," and which Gulton, the original manufacturer of the audio-sound source used in these games calls a "ceramic audio tone transducer." A thin disk of piezoceramic material, which changes its diameter when an electrical signal is applied across its surface, is bonded to a thin metal disk about an inch in diameter; total thickness is about half a millimeter. The metal disk acts as a restraining spring force on one surface of the ceramic, so that when the diameter of the piezoceramic disk changes, the bonded assembly will bend from a flat shape into a convex or concave shape, depending on the polarity of the signal. If the applied signal has a frequency of 2 kHz, the metal disk will vibrate and produce sound at a 2-kHz frequency.

Gulton recommends the CATT as an economical, low-profile sound source for applications such as electronic games, smoke detectors, appliance timers, alarm clocks and wristwatches, intrusion alarms, keyboard signalers, industrial controls, and personalized radio "beepers." Similar transducers are used in sonar devices and ultrasonic cleaners.

The audio-tone transducer operates at frequencies of 400 or 500 Hz up to about 4500 Hz. In Auto Race, the simulated engine sound increases in frequency and volume as speed increases (as you move from first to

Auto Race

second to third to fourth gear). When an oncoming car hits yours, a "bleep" is heard; if you complete the four laps in 99 seconds or less, a yodeling "bloop-bleep-bloop-bleep" is emitted; and if you can't make the four in 99 seconds, the "sound of defeat" is a short, low tone.

As the missiles in Missile Attack head toward "Your City," a V-2 type of buzz is heard; when you launch your anti-missile missile, you hear a bleep, and if you intercept the missile, the sound is a flat "blaah." If you successfully defend Your City against the 20 missiles, the final sound is an upward-spiralling All Clear; if an enemy missile hits Your City, the game is over, and you hear, loud and clear, the first three notes of "Taps"....

The basic sound in Football is a ticking clock. When the running back is clobbered, the "referee's whistle" sounds, like a short high chirp. After you press the Kick key, and if the computer decides to punt, two "whistle" sounds are heard; if it kicks a field goal (which is seldom), you hear a "charge" sound, something like an octave arpeggio in a minor key.

The tone produced by the "piezoceramic speaker" is sufficiently loud and clear that it could easily be used in some sort of inexpensive electronic musical instrument. Research may

Football

already be underway in this area, but toy companies keep future plans as secret as the military.

Toy Time

None of the three Mattel toys runs in real time. After all, a real football game has four 15-minute quarters, and that's only for the time the ball is actually in play. To keep Football down to a realistic playing time, the four quarters take up only 10 minutes of "ball"-in-play time.

An Auto Race minute is actually 50 seconds long, so the maximum "99 seconds" is actually only 82 seconds, although the time seems a lot longer if you're sweating out a tough race.

Missile Attack can last from 2 seconds (if the first missile is aimed directly at your city and you don't shoot it down) to about 40 seconds (if you shoot down all of the missiles aimed at your city).

Observations

If there's any doubt in your mind that the microcomputer actually keeps track of where you are, turn on Auto Race and stay in one lane. No matter which lane you pick, every one of the oncoming cars will aim directly for you. Rather disconcerting, what? And in Football, as you shift left and right, the defense will move in the same direction. The 20 incoming missiles in Missile Attack seem to take the same tracks each time, but — and this may be only my imagination — if you count on it, there will be a slight change or two, just enough to cause, for instance, two missiles to take entirely new directions, one in a straight line, the other, following directly after, will swerve at the last minute and drop right onto your city. So although the higher in the "sky" you intercept an incoming missile, the more points you earn, if you try too soon to shoot down a missile that changes directions, you'll waste a shot and endanger your city. Because if you miss, the computer causes a delay in arming a new anti-missile missile. Tricky.

One thing the microcomputer in Auto Race does not do is to make the opposition tougher as you get near the end of the lap (top of the LED stick). It just *seems* that way, because you have less reaction time as the distance between your car and the newly-appearing collision cars shortens. At each gear-speed, there's a minimum time you can do the four laps in, assuming no collisions. There's a slight spread in this minimum time, because some time seems to be lost in shifting lanes. For instance, running only in first gear, my best time ran between 87 and 91 "seconds"; in second, 56 to 60; in third, 36 to 38. As for fourth gear, I've never been able to make the four laps in

fourth without hitting several oncoming cars, so I don't know what the minimum time in high gear is.

The best strategy for Auto Race might be to take most of the lap in fourth, and then as you get near the end of the lap, switch to third, so as to have more reaction time. The catch is that shifting the gears may be just enough of a distraction to let your car get hit, and only a few hits will shove it right back to the starting line. All three games require total concentration; you can't look away for a second, or let your mind wander.

Auto Race has a chart that rates your time for four laps as "Under 30 seconds — World Champion Drive" down through "45-55 seconds — Showing Potential" to "75 or more seconds — Leave Car in Garage."

The two players of Football have a scoreboard to show who's winning.

Missile Attack rates under 25 as "U BLEW IT!", 25-50 as "THAT'S BETTER,", 50-75 as "GOOD" and 75 and over as "EXPERT MARKSMAN!" According to the manual, it's possible to score over 100 points, which then produces a special character in the first digit of the readout to show this special scoring condition. However, the best I've been able to do is 50, because at least a third of the 20 missiles shift their trajectories in flight, which makes shooting them down quite tricky.

Incidentally, when the battery weakens, the games just don't stop functioning. All three game manuals carry a similar notice; as Auto Race puts it: "If the blips of light representing cars or the Digital Timer appear to be malfunctioning, this is the first sign of battery wear. A fresh battery should solve the problem."

To prevent owners from opening a game and toying with the insides, special case screws are used; they can be removed only with a triangular-head screwdriver.

The Bottom Line

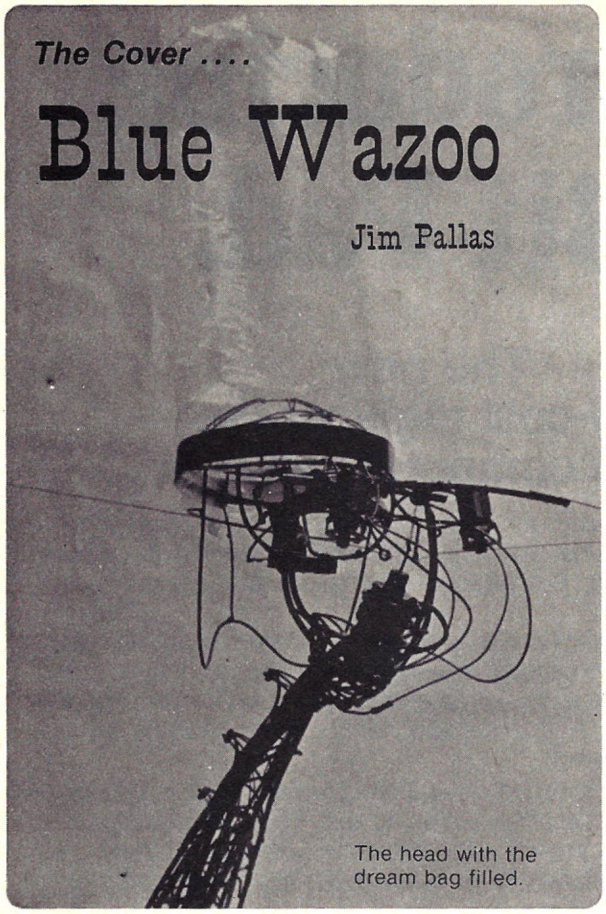
Auto Race and Missile Attack are priced between $20 and $25; Football runs from $30 to $35. If competing companies develop similar microprocessor-based toys, these prices could go down, but the trend, at least as of this writing, seems to be to develop more complex games. There are now at least two that play backgammon, and one that plays bridge. As for what new smart electronic games are going to be available by the end of the year, just anything a microprocessor chip can do, and enough people might buy, is probably on the drafting boards right now.

In the next issue, we'll take a look at the Parker Brothers nautical game, Code Name: Sector, which pits submarines against destroyers. ■

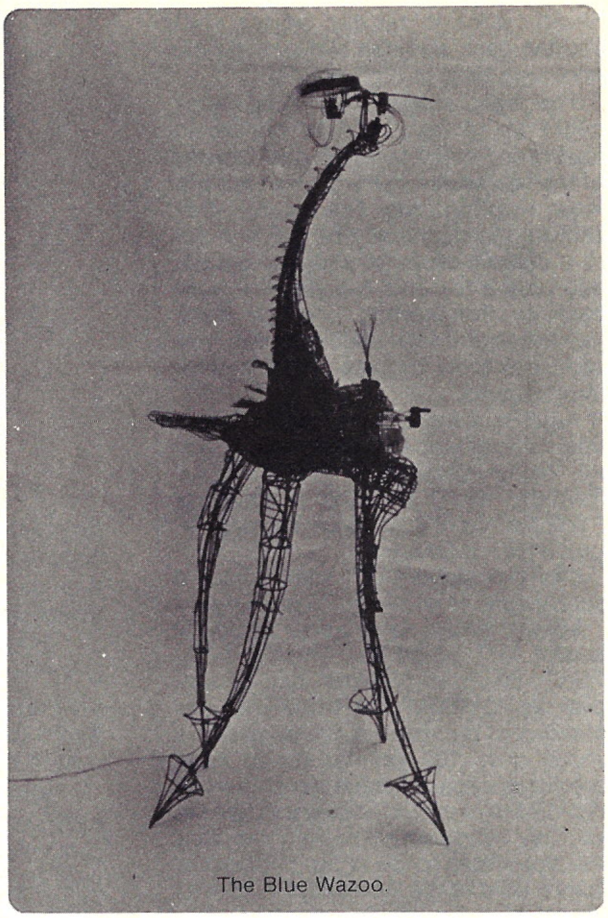
The Cover ....

Blue Wazoo

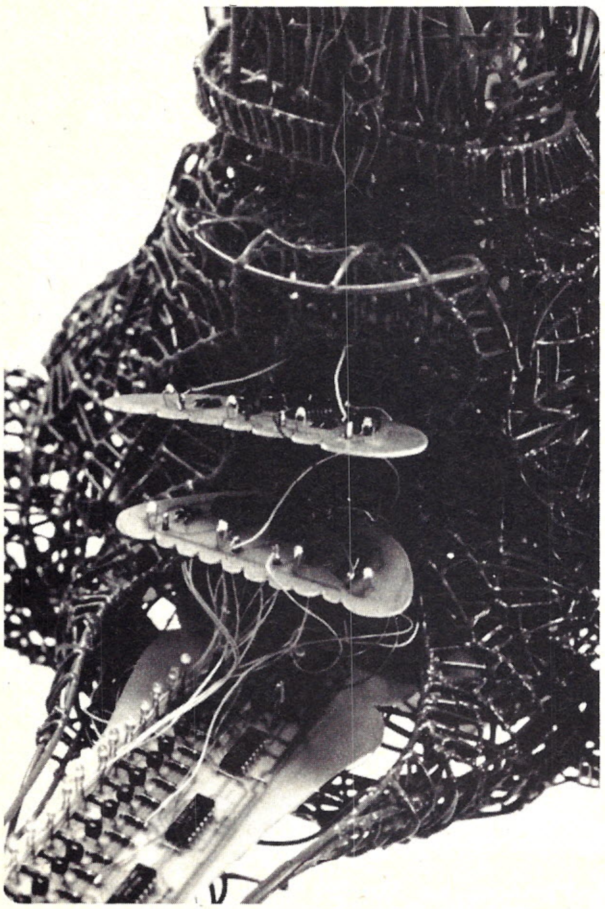
Jim Pallas



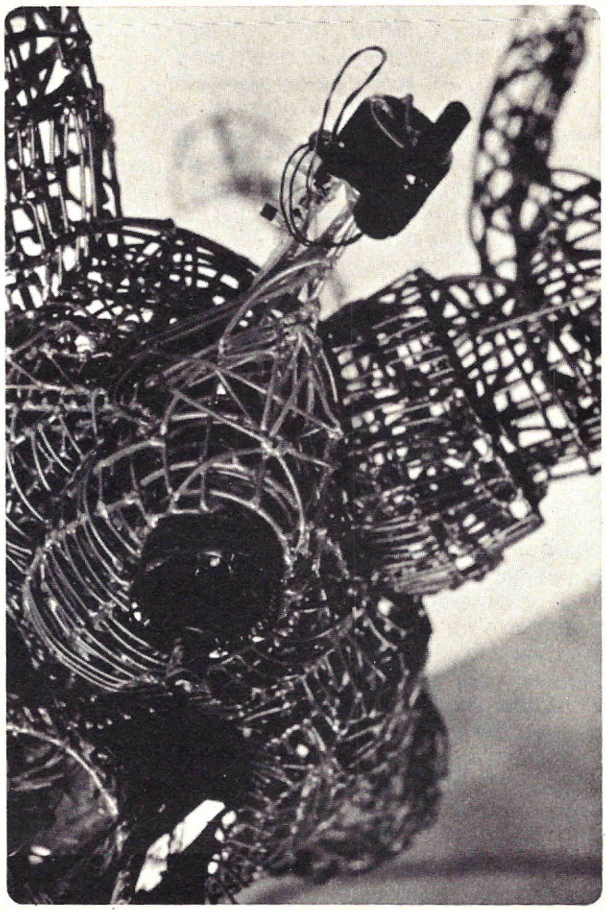
The head with the dream bag filled.



The Blue Wazoo.

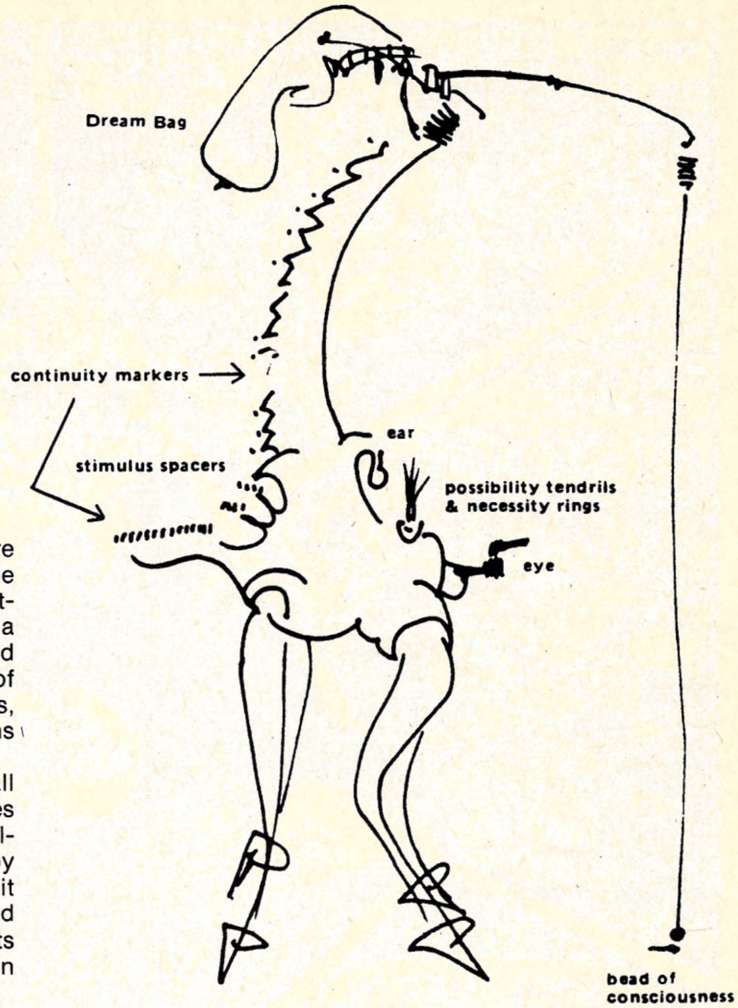


Continuity markers and stimulus spacers.



Necessity rings (center) and the eye (bottom).

```
      BLUE WAZOO

THE BEAD OF CONSCIOUSNESS
SUSPENDS CONTINUITY MARKING.
SPACED STIMULI.
BUDDAH-NATURE WAVERS TWITCH.
THE DREAM BAG FILLS.
POSSIBILITY TENDRILS
AND NECESSITY RINGS
POP UP.

BEAD DESCENDS.
```



The Blue Wazoo is primarily a welded steel structure covered with several coats of acrylic lacquer. The structure contains plastic shapes, ciruitry, wires, light-emitting diodes, solenoids, a motor, cloth, horsehair, a feather and a bead. The Blue Wazoo senses light and sound and responds with a behavioral repertoire of various LED patterns, movements, inflations, deflations, whirrs, clicks and jiggles. It is six feet high and weighs about 25 pounds.

Ambient light falling on the photocell (in the small velcro-mounted "gun" at the front of the body) increases the clock frequency (average: 2 hertz), of the 16 bit serial-in serial-out shift register. Data for the shift is generated by a microphone (at the front base of the neck). An 8-bit binary counter counts the output of the shift. Four nand gates are made conditional on data from various outputs of the shift and counters. Each nand gate controls an activity of the Blue Wazoo.

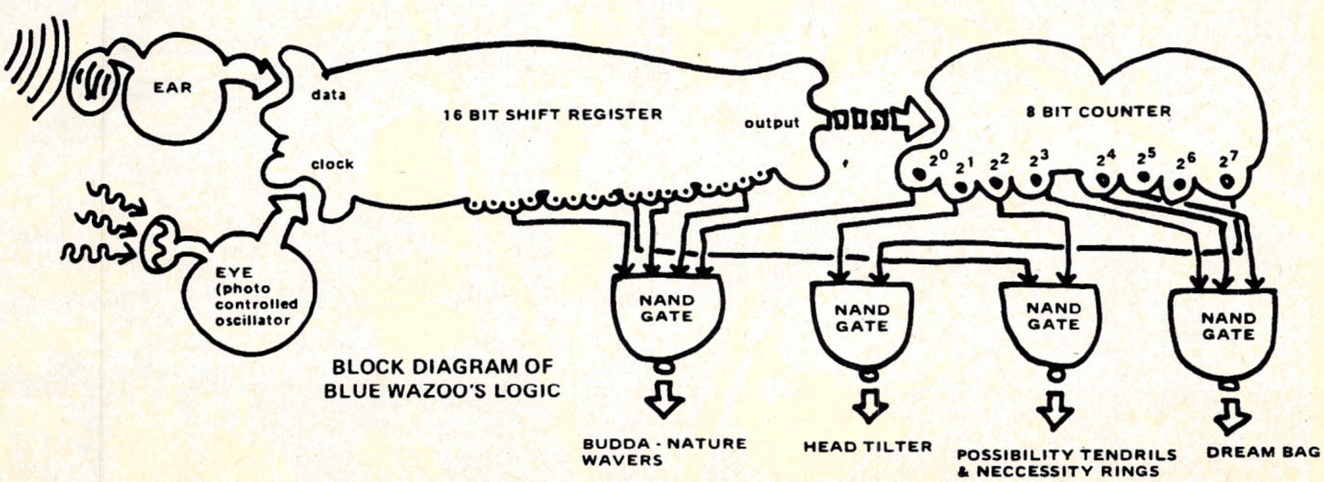
Visitors sense that the Blue Wazoo is reacting to something they are doing but they don't know exactly what. This often leads to superstitious behavior on their part. The artist is particularly interested in this cybernetic aspect of the work and feels that the dynamic interaction between viewer and artwork is one of the more exciting potentials in the use of the new technology in art.

The internal workings of a circuit and the resultant activities take on symbolic content as indicated by the individual names given to various parts of the sculpture.

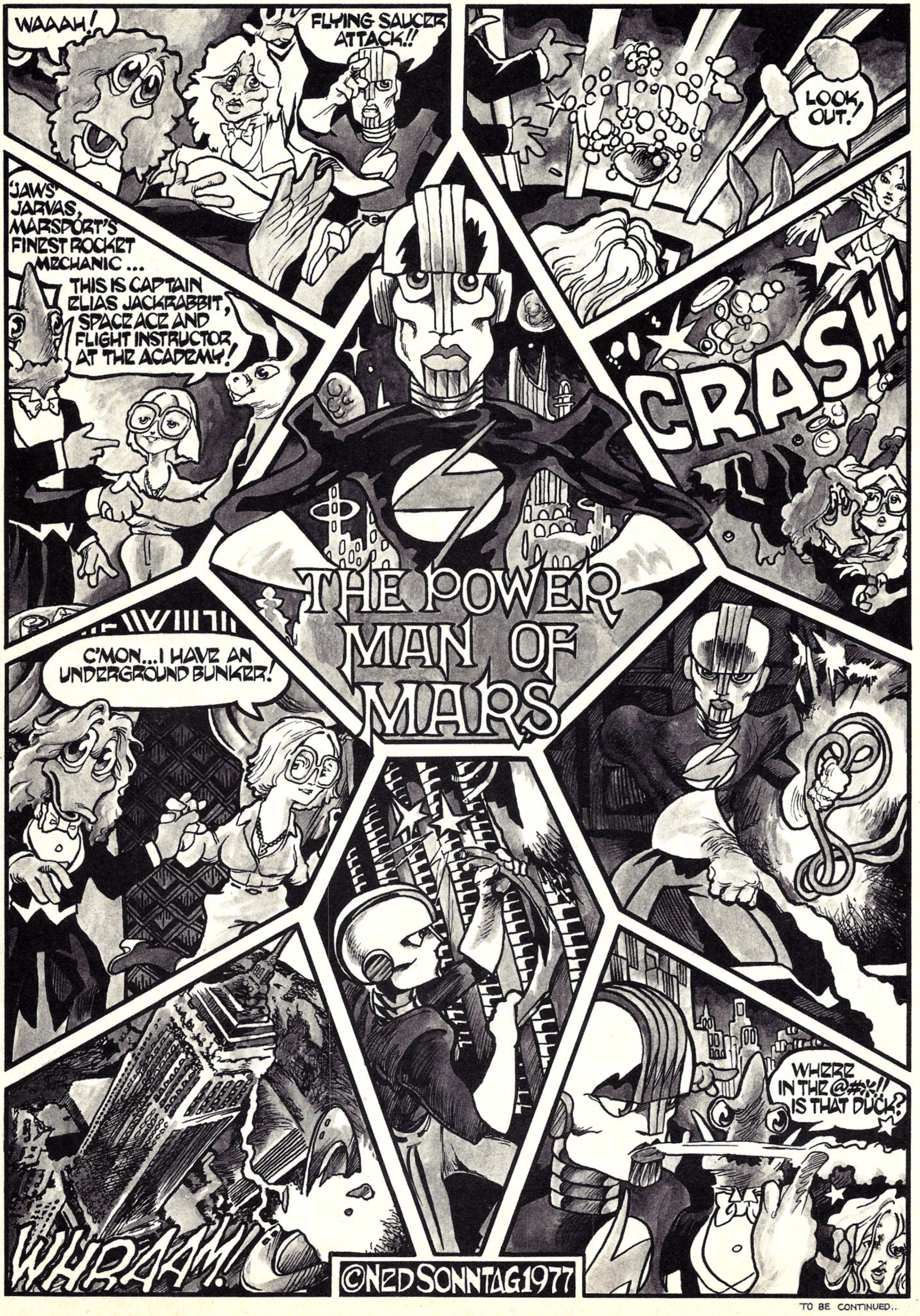
The Blue Wazoo was made by Jim Pallas in 1975-76 and is in the collection of the Allan Stone Gallery, NYC.

This article is appearing simultaneously in *Page*, the publication of The Computer Arts Society.

The artist, Jim Pallas is co-ordinator of the Detroit Art Works (they have a really fabulous T-Shirt with one of Jim's works on the front and Detroit Art Works emblem on the back. Black shirt, adult sizes $7.00 postpaid. Well worth it — DHA). Jim is also on the faculty of Macomb County Community College. He can be reached at 1311 Bishop, Grosse Point Park, Michigan 48230. ■

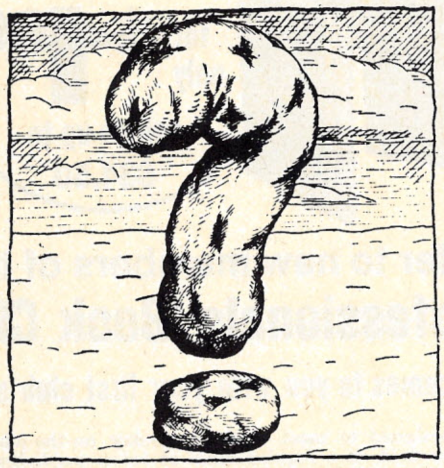


BLOCK DIAGRAM OF BLUE WAZOO'S LOGIC



TO BE CONTINUED..

Five Who Most Disturbed The Thought of Man



COPERNICUS, 1473-1543,
who deprived man of his earth-centered universe.

References:

Kuhn, Thomas S., *The Copernican Revolution,* New York, R. R. Bowker Company.

DARWIN, 1809-1882,
who deprived man of his status as a special creation.

References:

Darwin, Charles: Evolution and Natural Selection, Loewenberg, Bert James (Editor), New York, R. R. Bowker Company.

Darwin, Charles — Autobiography and Selected Letters, (Edited by Francis Darwin), New York, Dover Publications, Inc.

Hofstadter, Richard, *Social Darwinism in American Thought,* Boston, Mass., The Beacon Press, Inc.

FREUD, 1856-1939,
who deprived man of belief in rational self-determination.

References:

Freud, Sigmund, *Delusion and Dream,* Boston, Mass., The Beacon Press, Inc.

Freud, An Autobiographical Study (Translation by James Strachey) (2nd ed.) London, England, The Hogarth Press, 1946.

Jones, Ernest, *The Life and Work of Sigmund Freud,* New York, Basic Books, Inc.

EINSTEIN, 1879-1955,
who deprived man of confidence in his own observations of the physical world and forced him to understanding, as distinguished from proof only, of phenomena.

References:

Barnett, Lincoln, *The Universe and Dr. Einstein,* New York, R. R. Bowker Company.

Coleman, James A., Relativity for the Layman, New York, The New American Library, Signet, P2049, 4th printing, January 1962.

Lovell, A. C. B., *The Individual and the Universe,* New York, R. R. Bowker Company.

Gamow, George, *Biography of Physics,* New York, Harper and Brothers, 1961.

Rogers, Eric M., *Physics for the Inquiring Mind,* Princeton, NJ, 1960, Princeton University Press.

COMPUTER, 1948-?
who deprived man of his unique position as an intelligent manipulator of his environment and creative solver of complex problems.

References:

Feigenbaum, Edward A. and Feldman, Julian (Eds.), *Computers and Thought,* New York, 1963, McGraw-Hill.

Rothman, Stanley and Mosmann, Charles, *Computers and Society,* Chicago, 1972, Science Research Associates.

Spencer, Donald D., *Computers in Society,* Rochelle Park, NJ, 1974, Hayden Book Co.

Brown, Les and Marks, Sema, *Electric Media,* New York, 1974, Harcourt Brace Jovanovich.

Nelson, Theodor H., *Computer Lib/Dream Machines,* Chicago, 1974, Hugo's Book Service.

Wessel, Milton, *Freedom's Edge: The Computer Threat to Society,* 1974, Addison-Wesley.

Weizenbaum, Joseph, *Computer Power and Human Reason,* 1976, W.H. Freeman. ■

A Creative Computing Equipment Profile...

Radio Shack TRS-80

Stephen B. Gray

You open the large box and take out three smaller boxes. The first contains a 12-inch video display monitor; the second, a combination of ASCII keyboard, CPU, memory and cassette interface, all in one molded case, plus a separate AC power supply; the third contains a standard Radio Shack CTR-41 portable cassette recorder. Also included are several cables and two cassettes, one blank, the other containing two games: blackjack and backgammon.

Setup is simple. Three 5-pin DIN connectors plug into the keyboard case: one from the monitor, another from the power supply, the third from the cassette recorder (needed only if you plan to record programs or load taped programs into the TRS-80). Three small plugs connect to the CTR-41: into the EAR, AUX, and REMOTE jacks. (A dummy plug in the MIC jack disconnects the built-in microphone to prevent picking up ambient sounds while you load tapes.) Plug in the power cord and you're ready to go. Incidentally, although the three DIN plugs are identical at the metal end, the plastic handles are all differently shaped, and the keyboard case is stamped TAPE, VIDEO, and POWER. But even if you plug one of these connectors into the wrong jack, you won't blow out anything; the designers have made the TRS-80 as idiot-proof as possible. If you'll be plugging and unplugging your TRS-80 frequently, you may want to label the three keyboard-case plugs, as well as the three cassette-recorder plugs, for faster setup time.

Turning On

Turn on the monitor by pressing the red Power button. Turn on the keyboard by pressing the small ON button at the back of the case. A red LED near the lower right end of the keyboard will light up to remind you that the computer is on. The monitor displays

```
READY
>_
```

The READY means the computer is waiting for your input; the "greater than" sign is the "prompt," which means the computer is ready for a line number (or other command), and the underline sign is the cursor, which shows where your input will be placed on the screen.

The TRS-80 has two operating modes. The simplest is "calculator mode," without line numbers, which will calculate any expression you type after PRINT. If you type PRINT 5-2 and press the ENTER key, a 3 will show up on the next line. Typing PRINT 7/22 will display .318182; the maximum is six decimal digits. If you type PRINT 7/222, the result is 3.1531E-02 instead of .031531; for numbers less than .1, the computer prints out in scientific notation.

Errors

To correct a typing error, press the backspace key (←) as many times as necessary to erase the error. Press CLEAR to blank the screen.

In computer mode, use line numbers to write programs. An END statement isn't required; you need type only RUN.

Three error messages signal various problems. If WHAT? is displayed, the computer does not understand the instruction. Perhaps you typed PRNT when you meant PRINT, or left out a closing parenthesis and typed PRINT 2/(4-3 instead of PRINT 2/(4-3). If HOW? is shown, the computer understands the instruction, but can't execute it, such as if you use GOTO with no line number to branch to, or use a line number beyond 32,767, or try to divide by zero. If SORRY comes up on the screen, the computer has run out of memory.

Basic TRS-80: keyboard/computer, video monitor, cassette unit, power supply.

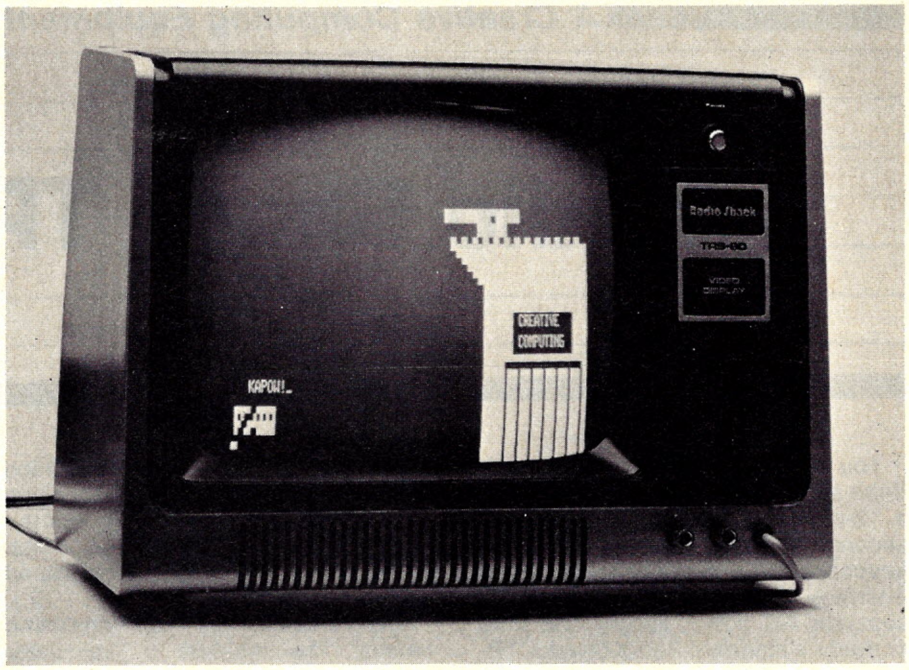
Radio Shack BASIC

According to the computer's designer, Steve Leininger, TRS-80 BASIC "was designed around Palo Alto Tiny BASIC," with the addition of floating-point math. two strings (A$ and B$), and "some other things." The $599.95 price for the complete TRS-80 system includes Level-I BASIC, stored in a 4K ROM. Level I includes four commands (LIST, RUN, CONT, NEW), fifteen statements (REM, LET, FOR-NEXT-STEP, GOSUB-RETURN, STOP, END, GOTO, IF-THEN, INPUT, ON...GOTO, ON...GOSUB, PRINT, DATA, READ, RESTORE), three functions (ABS, INT, RND), ten math operators (add, subtract, multiply, divide, equals, not equal to, and four greater-than or less-than operators), two cassette-memory commands (CSAVE, CLOAD), and several special commands. These last include graphics commands such as SET, which turns on a rectangle (a little larger than a hole in an IBM card) at the designated XY coordinates; RESET, which turns off a graphics rectangle; and POINT, which checks for the presence of a rectangle at the specified XY coordinate (if on, returns a 1; if off, returns a 0 — useful in writing games). CLS clears the screen. MEM displays the number of bytes left in memory.

The variables are A through Z, up to six significant digits; A(n), four bytes per subscript, from 0 to MEM/4; A$ and B$, character strings up to 16 characters. A$ and B$ cannot be used logically; that is, A$ can't be compared with B$.

The RND is different from the random function used in most BASICs; instead of providing a decimal number between 0 and 1, it provides, if RND(10) is used, for example, random integers between 1 and 10, inclusive. Successive RUNs provide different sets of random integers each time.

Most of the statements and com-



TRS-80 graphics: cannonballs demolish the fort, accompanied by noise-words such as KAPOW!

mands can be abbreviated, usually to one letter and a period. Since TRS-80 BASIC permits multiple-statement lines, this feature permits making the most of the 4K dynamic RAM memory; also, neither LET nor END is required.

For printing, the screen will display 16 lines of 64 characters; upper case only. For graphics, the screen is divided into 1024 rectangles, 128 horizontally by 48 vertically. Graphics and text may be interspersed. The PRINT AT function will print any message starting at the indicated rectangle. For example,

PRINT AT 640, "HELLO"

will print HELLO at the beginning of the line about two-thirds of the way down the screen.

Cassette Recording

Data recording using the CTR-41 is about as simple as possible. To record a program from the TRS-80, press the recorder's RECORD and PLAY keys simultaneously, then type CSAVE on the keyboard, and ENTER this command. When the program has been recorded (at 250 baud), the TRS-80 will automatically turn off the recorder and flash READY on the screen. According to the manual, "many computer users make a second or even a third recording of the tape, just to be sure they have a good recording." The half-dozen recordings I made all recorded and played back perfectly, without the need for additional recordings.

To play a recorded tape, set the CTR-41 VOLUME control to between 7 and 8, press the PLAY key, type CLOAD on the keyboard, and ENTER. This starts the tape motion. An * will appear at the top line of the monitor screen; a second * next to the first asterisk will blink, indicating that the program is loading. When loading is done, the CTR-41 turns off and the screen indicates READY. All you have to do is type RUN and ENTER.

The tape-recording format is said to be similar to the format described by Hal Chamberlin in his now-defunct publication, "The Computer Hobbyist."

Incidentally, the computer knows when to stop when it's reading a program from cassette because, when the computer wrote the program, it automatically wrote a header that indicates how many bytes the program uses, and also gives the starting location at which the program will be loaded in computer memory.

Blackjack, Backgammon

A "game package" consisting of blackjack and backgammon on a cassette is included with the TRS-80. When I loaded blackjack (which took a little over a minute and a half) and tried to play it, the first part of the program ran properly, but then the screen filled with a meaningless collection of alphanumerics, and the game halted. Seemed like a dead end, but then I remembered the line in the manual about a second or third recording, and tried another CLOAD on the same tape, after first wiping out the bad program with a NEW. The game ran perfectly. Then I remembered that during the first CLOAD, the second asterisk had blinked only a couple of times, and then stayed on. That should have been a clue. Anyway, I played Blackjack, and won the first six games. Then I started to lose, eventually lost all my winnings. Was this Blackjack actually programmed to let me win the first few games, and lead me up the garden path?

Then I loaded the Backgammon game. The accompanying booklet says "If you don't know the rules, you can pick up a copy at any bookstore or you can play the computer game a couple of times and the rules will become obvious. The computer will not let you make an invalid move." The computer didn't, but I couldn't figure out the end game anyway. I do wish they'd taken a couple more pages and given the rules.

Using the TRS-80 in the lab to compute theoretical test results.

Incidentally, the erase-protect tabs on this cassette are both broken off, to prevent your accidentally erasing either game.

User's Manual

As of this writing, the 230-page User's manual wasn't quite ready, so much of this article is based on the 30-page Preliminary User's Manual, adequate but not detailed, and with no attempt made to teach BASIC, which presumably the larger manual does.

Comments

The microprocessor in the TRS-80 is a Z-80, although a great many users won't really care if it's a 4004, as long as the computer performs. And it does, very well. Operation couldn't have been made simpler; by having BASIC in ROM, all you do is plug in the machine, turn it on, and start programming.

Some readers are bound to ask, why not use a standard TV set instead of a monitor? Because with a monitor you get a higher resolution, and the sharply-focused characters are much easier to read. If you were to try using a standard TV set, the characters would smear rather badly. The problem is one of bandwidth; the signal coming out of the TRS-80 is over 5 mHz, and most TV sets are limited to 4.5 mHz, or even 3.5 mHz. However, with the forthcoming Level II BASIC, you can switch from 64 to 32 characters per line, which widens the characters and allows them to look quite good on a standard TV. Then all you need is an RF modulator such as a Pixe-Verter, and you can use your own TV set.

One peculiarity: if you type PRINT X, without having previously assigned a value to X, the result will be 965. Every time. PRINT B provides -.5, and PRINT J gives you 8.18799. The explanation is quite simple: LEVEL I BASIC doesn't clear out the variables. LEVEL II BASIC will do this. Meantime, follow the instructions given in just about every book written on BASIC; initialize your variables.

(The reason PRINT X provides 965 seems to be that, because semiconductor memory locations can't be made exactly alike electronically, the individual bit positions thus have a built-in tendency to assume either the 0 or 1 state when the computer is turned on. This tendency can apply to a group of locations, so that the same value could come up when entering anything from PRINT M to PRINT Q, for example. And if such a variety of values exists in a single computer's memory, it must follow that no two TRS-80 computers will print out the same set of values for variables not previously initialized. And if no two sets of these variables are alike, then every TRS-80 memory is as unique in this respect as its owner's thumbprint.)

The Future

Under a cover at the left rear of the keyboard case is a 40-pin bus for connecting peripherals and expansion memory. Level II BASIC, which is 8K Extended BASIC, will be available early this year, and replaces the single Level I ROM with two ROMs. LEVEL II adds the trig functions, exponentiation, SQR, and various other goodies found, for the most part, in MITS Extended BASIC. Actually, the next hardware to be available will be an additional 12K of RAM, and a printer. More about those at a later date, along with details on future add-ons such as a serial interface (so the TRS-80 can be used as a terminal), a second cassette interface, an assembler/editor, a disk operating system and mini floppy-disk storage system, and many others, including an adapter to permit using S-100 boards with the TRS-80.

In the next issue, we'll discuss the 230-page manual, and the four applications programs now available (Payroll, Education—Math 1, Kitchen, and Personal Finance) for the Radio Shack TRS-80 computer system. ■

Heath H-8 System

John Lees (Hardware) and Steve North (Software)

[*Ed. Note: This is a review of three Heath computer products assembled by Heath employees as part of the manufacturer's pilot-run quality-control program. At a later date, when the kits of these three products are available, we will review them, as well as the H-11 computer.*]

Heath computers, after a year of rumours and speculation, finally arrived at the *Creative Computing* building last month. We unpacked the wired H-8 8080-based computer, H-9 video terminal, and H-10 paper-tape reader/-punch with high expectations and eagerly set in on our review of the system.

The Hardware

Unfortunately we could not begin right away because we could not get the H-9 video terminal to work, and Heath had forgotten to include the manuals. When the manuals did arrive, we traced the problem to a broken wire and fixed it easily.

Meanwhile we decided to check out the H-8 computer as far as we could without a terminal. Heath had included a parallel I/O card separately and we first installed it in the mainframe before proceeding. Our initial test went like this:

John: "Plug it in."

Steve: "Okay."

John: "Turn it on."

Click.

John: "What's that!?"

Steve: "Look!"

John: "Turn it off! Turn it off!"

Silence.

John: "Where's the smoke coming from?"

Steve: "Back there ... must be the power supply...."

John: "Yeah. The fuse is blown."

After installing a couple of Radio Shack diodes in our Heathkit computer, we gingerly tried again. This time nothing went wrong and everything seemed to work correctly. Temporarily baffled, we eventually discovered that the board connectors used by Heath did not prevent plugging in a board out of line. If a board is plugged in offset by one pin up, the -18 volt supply is shorted to ground, which is what we must have done.

Talking with Heath revealed that they now glue a plastic barrier on each board which prevents this — obviously an afterthought. We are not overly impressed with the sturdiness of the connectors, the difficulty with removing boards (there are a number of screws that must be removed), and the difficulty of taking out the motherboard to repair the 18-volt supply. If you're one of those hobbyists who continually yank boards in and out of your computer, the H-8 is *not* for you. On the other hand, if you intend to get your computer working and then leave it alone, the method of securing the boards is superior. A definite plus if you move your computer about a lot.

At this point we want to thank Heath for one very good feature. All 120-volt connections are completely enclosed in a metal box. You can't accidentally electrocute yourself while poking around the H-8 as you can with so many other machines. Most manufactures don't bother with UL approval. Heath does. Bravo! This goes for the H-9 and H-10, also.

We then proceeded to check out the H-8 with the short program included in the manual. Entering it through the pushbutton front panel was nicer than using toggle switches, although we don't understand why an octal display is used for an 8-bit machine. Upon executing the test program, the display reassuringly told us that "YOUR H-8 IS UP AND RUNNING." We set the H-8 aside until we had a terminal up and turned our attention to the H-10 paper tape reader/punch.

Heath H-8 8-bit computer

Without the rest of the system, all we could do with the H-10 was to copy a tape, which one does by pressing a button on the rear of the H-10. One-third of the way through copying a 1000-foot tape, we discovered the first serious drawback to the H-10. Somehow the tape being read had become twisted and folded in the reader, and we had the devil's own time clearing it. There is no easy way to clear such a problem tape. You can't simply lift it out of the reader as you can with a Teletype, because the tape goes through several closed apertures and fits closely between the lamp and photoresistors, and under two spring clips. But we got it cleared and proceeded.

However, another 100 feet further on, the punch began making a strange sound and the punched tape fell out, chewed in half. Examination showed that one solenoid was permanently "on," blocking the tape path. Oh, well. Then it dawned on us that we were halfway through a 1000-foot roll with no way to clear it from the reader. So we pulled 500 feet of tape through the reader, saying nasties about Heath the meanwhile, pulling it being faster than feeding it. Replacing a power transistor cured the problem, but we decided not to push our luck by duplicating another long tape. Additional bad points are the chad box (a joke) and the reader's intolerance for wrinkled tape. The H-10 is definitely a *hobby* peripheral. Don't be fooled by its good looks into thinking it can take a commercial pounding.

What else to say about the hardware? Of course Heath uses a unique 50-pin bus, and unique con-

nectors to go between peripherals. Actually, the connectors used are much sturdier and easier to solder to than the 22-pin EIA standard, and the 50-pin bus makes a lot of sense from an engineering standpoint. If you're going to have an all-Heath system, you've nothing to worry about, and if not, Heath-compatible boards and adaptors will certainly show up in the near future.

The I/O boards in the H-9 and H-8 have many, many jumpers that allow you to select baud rates and stop bits and hand-shaking, etc., to match whatever you're using. But such changes require a soldering iron. Using Heath peripherals with several different computers will be a real pain unless you install some handy switches for yourself. Again, no problem in an all-Heath system, and it would have upped the cost for Heath to include such convenience features.

The H-10 can be used with a parallel I/O card, which allows it to run at its own speed, or the H-10 can be plugged into the H-9 terminal, in which case the combination runs at 110 baud, emulating a Teletype. The H-9 displays 12 lines of 80 characters and Heath obviously intends to offer an expansion to 24 lines. The H-9 also has a plot mode, which looks interesting but which we didn't have time to try out.

Manuals and Documentation

As expected, the Heath manuals are far, far superior to anything any other manufacturer makes available. The circuit descriptions and troubleshooting guides are excellent. The manuals, coupled with the ability to call Heath for assistance, make the Heathkit a clear first choice if you desire to build your hobby computer system from a kit, yet are not yourself an experienced technician.

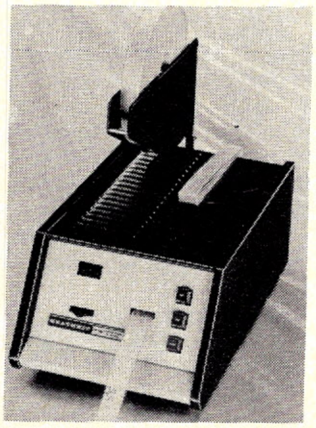
The Software

PAM-8: The Heathkit H-8 comes with a front-panel monitor program contained in non-volatile Read-Only Memory. This monitor accepts commands from the front-panel keypad to control the computer, and also manipulates the display. The front panel consists of sixteen pushbuttons and eleven seven-segment displays. The H-8 front panel does not have direct control over the machine. The PAM-8 permits examination and modification of memory, CPU registers, and I/O ports, single-stepping, and loading and saving of memory with a peripheral device. PAM-8 supports a realtime clock and also permits you to reset the system (re-enter PAM-8) by pressing two buttons on the keypad simultaneously.

Since it is not a direct display of the internal state of the machine, the H-8 front panel is not as well-suited for hardware debugging as the front panel on an IMSAI. On the other hand, it does offer many convenient features for people who prefer to use their computers, not debug them. To save a program on the device at port 370 octal, enter a few digits and press DUMP. To load a program from the boot device (at 370 octal) just press LOAD and sit back. With a "dumb" front panel, you would have to enter a long I/O driver program. With the H-8 front panel, you can examine memory in a normal mode, examining a particular address and stepping forward and backward (with the + or - keys on the keypad), or use the "alter" mode, in which you enter consecutive bytes of memory.

Heath H-9 CRT keyboard terminal

The H-8 front panel use octal notation. While Heath may have felt that this was logical, since the 8080 instruction set is octal, there is a strong predominance of the use of hex in the microcomputer field. One byte just doesn't conveniently divide up into groups of three bits. It would also have been nice if Heath had built a little more flexibility into PAM-8, so that the system could be booted up from a device other than the one at 370 octal. However, front panels in general are becoming less and less popular as people discover that it is impossible to do anything significant without a terminal. Therefore, it is often more practical to include a PROM monitor that talks to a terminal, since you generally ends up with one. It is also tough to understand why Heath decided to use the 8080 instead of the Zilog Z-80 microprocessor. While we suspect that part of the attraction of the Z-80 stems from the fact that it begins with a 'Z', it is nevertheless an obvious choice over the 8080 in designing a new product, since its instruction set is more sophisticated than the humble 8080's, and use of the Z-80 instead of the 8080 also generally reduces the package count. PAM-8 resides in low memory in the H-8. This will make it difficult to use most of the 8080 software available in object form, since most software is assembled to run at zero. In most other micros, PROM memory is located way up at the top of the addressable memory, thereby not interfering with anything you want to load in low memory. It is safe to say that few people will be using MITS BASIC with their H-8's.



Heath H-10 paper-tape reader/punch

BUG-8: BUG-8 is a debugger for 8080 object programs. BUG-8 permits your general debugger functions (examination and modification of memory, registers, single stepping, etc.) with a terminal. This is definitely a handy thing to have if you do much assembly language programming.

TED-8: To create assembly-language programs and general text, Heath supplies TED-8, a text editor. This editor is pointer-oriented, and permits listing of the text, insertion/replacement/deletion, searching for a given character string, etc. Although your first impression at looking at the TED-8 documentation is that it is difficult to learn, it is actually rather straightforward. Since all commands have the same basic format, one need only plug in the correct parameters to perform a specific function. Unfortunately, in learning to use TED-8 we did not use this format a few times, and managed to crash TED-8. TED-8 also permits you to save and load text information from an external device.

HASL-8: While the acronym for this software may have been rather poorly chosen, Heath's assembler seems to be a good 8080 assembler. Assembly is done from source code on cassette or papertape, rather than memory. While this method is slower than a completely in-core assembly, it also eliminates the need to have 32K of memory to assemble a well-commented 1K program. The bad news is that sometimes Heath does not use the normal Intel-approved mnemonics. No one else calls a JNZ (Jump if Not Zero) a JNE (Jump if Not Equal) (even if, after a compare instruction, you really are testing for

equivalence). There is no need to create incompatibility at the assembly-language level! What are a couple of characters, here or there? If, on the other hand, you never plan to use other people's programs, it doesn't make any difference. HASL-8 has conditional assembly features.

BASIC: Heath ("Benton Harbor") BASIC (8K and 12K) offers most of the features that have become standard for microcomputer BASICS: floating-point math, multiple statements per line, arrays, string scalars and arrays, a full function library, and so on. Heath 12K BASIC also has some special-purpose commands for manipulating the front-panel display and to read the front-panel keypad. Although Benton Harbor BASIC is not nearly as fast as MITS BASIC, it seems reliable and should serve the needs of Heathkit buyers. The fly in the BASIC ointment is a feature called command completion. As you type in statements, BASIC looks at the characters you enter, and if it thinks you are typing a keyword, it completes the keyword for you on the terminal, and apparently also condenses the keyword into a single byte in memory. For example, you type GOS and the machine finishes the word with UB to make GOSUB. So we found ourselves typing PRINT INT and RUN N quite often. A representative of Heath explained that after using the software for a while, you can anticipate when a keyword is coming up, and wait for the machine to complete it. In fact, he said it felt unusual to go back to a machine without the command completion. It's hard to decide whether this feature is actually desirable or whether it is a gimmick. Heath is rumoured to be considering a modification to permit you to use command completion or turn it off, which we would heartily recommend (this would make the decision as to the value of command completion an individual thing, rather than an academic question). The other not-very-handy thing is that when you do a string INPUT, the string must be enclosed in quotes. The reason for this, as far as we can guess, is that BASIC uses the regular line-input routine to read a character string, and the " is needed to turn off command completion. In Extended BASIC there is a LINE INPUT statement that works like a normal string Input, without command completion and without quotes.

Extended BASIC does have a nifty feature that automatically vectors you off to a particular BASIC subroutine when control-B is typed on the keyboard. A real-time clock is also available to BASIC. So, except for those two related features that are slightly annoying, Heath BASIC is a very adequate BASIC for the hobbyist.

It is commendable that Heath is providing cheap software for the H-8. This will tend to discourage software thievery by H-8 owners. Heath has nothing to fear from non-H-8 owners (not only will H-8 owners not be using MITS BASIC, but Altair owners will not be using Heath software, due to its dependence on PAM-8). Heath has also developed a reasonable standard for the format of the data recorded on cassette or papertape, rather than just throwing data out at the peripheral and then sometime later yanking in whatever goes by. Heath's design philosophy for software seems to have carried over from the philosophy behind their hardware: nothing flashy or innovative, but it is cheap and it works reasonably well. Neither portion of their product is aimed at the microcomputer fanatic (who by now has probably already bought a system anyway (which came first: the computer or the computer fanatic?). It is also not a product for the masses out there, unknowingly waiting for the phantom PET to be dropped upon them like some kind of bomb. Rather, it is ideal for the person who likes Heathkits and wants to get into personal computing Heath-style. ■

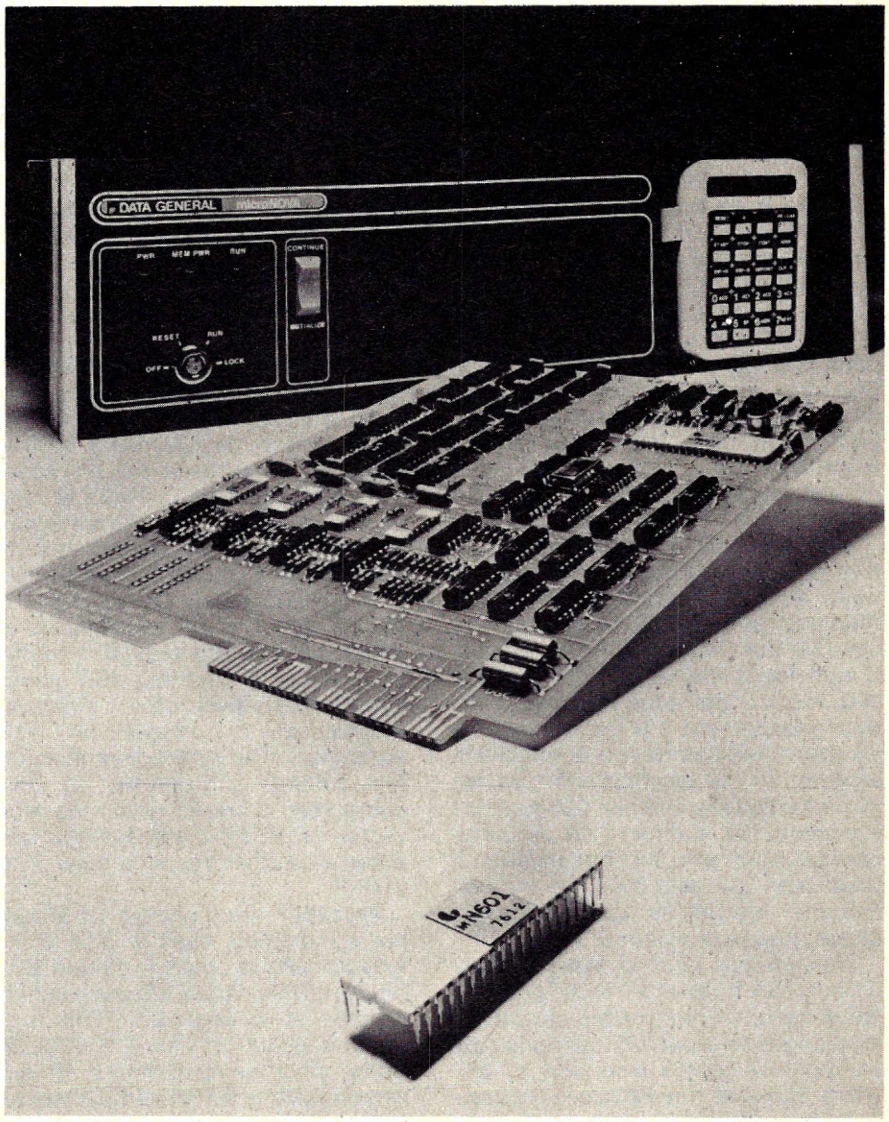
How Would You Like A 16-Bit Computer?

Webb Simmons

Those of us whose experience has been limited to the 8-bit microcomputer chips will be seeing some new and revolutionary computer features before very long. I say this because the prices on the 16-bit microcomputer chips are coming down and some, such as the Nova chip, are being second-sourced. Data General is the prime source for the Nova chip, but it is rumored that Fairchild has a Nova chip in the works for the near future.

Some of the new features will be mighty nice but some will be a little disappointing. An example of a disappointing feature is the lack of a *stack*. The classic PDP-8 did not use a stack, so neither does the Intersil imitation of the PDP-8. The Nova 1200 that Fairchild is copying did not use a stack so neither will their copy. At least the Nova copy will allow standard code to be placed in ROM (EPROM, etc.) whereas a PDP-8 standard subroutine cannot be placed into ROM because of its archaic subroutine linkage. The copy of the PDP-11 (currently the LSI-11, but perhaps others are to come) has a stack of course; its disappointing feature is primarily its high price.

The *new* features I will discuss are not new at all, as you might have guessed. They center around the use of memory locations to perform register-like functions. Most of the giant computers allow such shenanigans and all of the so-called minicomputers absolutely require them. They could not survive without these features (with the possible exception of the PDP-11). Minicomputers have been made with just *one* 8-bit register and the PDP-8 is not particularly fat. Several very successful minicomputers such as the HP-2100 by Hewlett-Packard have had only two registers.



Data General's microNOVA chip is featured in the one-board CPU with 8K bytes of RAM, and in the complete microNOVA with its detachable "console."

Prices of 16-bit microprocessor chips are coming down; some are now second-sourced. Here's a rundown on the *new* features you can expect from 16-bit machines in the near future.

The addressable memory increment in minicomputers can be 8-bit bytes (examples are the PDP-11 by DEC and CIP/2200 by Cincinnati Milacron) or 16-bit words (examples are the Nova series by Data General). These are not the only addressable increments; a few are 12 bits, 18 bits, etc., but let us not discuss these at all. Regardless of the addressable increment it is conventional to think of memory as *bytes* of 8 bits and *words* of 16 bits. A word on the PDP-11 must be two bytes starting with an even-numbered location, but a word on the CIP/2200 is two bytes, starting anywhere. A byte on the Nova is either the upper or lower half of a word.

The Nova minicomputer has features of most minis that are both good and bad when compared with existing 8-bit microcomputer chips and it is the chip you are most likely (big question mark here) to see, so let me use it as an example.

On the Nova, any 16-bit memory word can be used as a counter, either to control a loop or for any other purpose. The Nova uses the instructions ISZ (increment and skip if zero) and DSZ (decrement and skip if zero) to manage these words. If the addressed location does not become zero upon an ISZ (or DSZ), the next instruction (which might be a jump, JMP) is executed in regular sequence, or else the next instruction is skipped over.

The Nova allows any memory location (and this is always a 16-bit, 2-byte word) to be used as a *vector* or *pointer* (address pointer). Such a vector is used for the *indirect* addressing mode. If one loads *indirectly* from location 100, it is not the value of location 100 that is loaded but it is the value at the location whose address is contained in location 100. If location 100 contains the value 200 and location 200 contains the value 300, a direct (the ordinary) load from 100 gets 200, but an indirect load from 100 gets 300. The indirect memory references can be loads, stores, jumps or whatever.

The address field in Nova instructions is only 8 bits wide. This means the Nova cannot directly address without indexing more than 256 locations (256 words)—the words from 000 to 0FF (hex). However, the Nova can *indirectly* address these same 256 words, each of 16 bits, and each of these is able to address the entire machine. These first 256 words on the Nova are called "page zero." Page zero figures strongly on most minicomputers but in a very different sense than the page zero *restart* locations on the 8080 and Z-80.

All Nova instructions can also be indexed (whether or not they are indirect) in which case the 8-bit address field of the instruction is regarded as being a signed value from -128 to +127 with respect to AC2 of AC3 or the program counter. AC2 and AC3 are Nova registers. The Nova has four 16-bit registers: AC0, AC1, AC2 and AC3; all four can be used as accumulators but only AC2 and AC3 (or the program counter) can be used as an index register. Indirect addressing may be combined with indexing on the program counter to permit any nearby word to be reserved for use as an address vector. Indexed addressing on the program counter is *relative addressing* as it is defined for jumps on the Z-80.

The most common subroutine jump or call mechanism on hobbyist computers is the transfer to a location identified by a 16-bit address field with the return vector (return location) *pushed* onto the stack. This stack can be in memory as it is for the 8080 and Z-80, or it might be in the CPU itself as it is for the 8008 or IMP-16. Such hardware-managed stacks are relatively uncommon in minicomputers. The PDP-11 has a stack but the older Nova computers do not. On the Nova the return vector is always placed into AC3, destroying any value that might be there before executing a JSR (jump to subroutine). Even this subroutine mechanism is unusual for minicomputers. The most common method for calling subroutines in minicomputers is one in which the return vector is stored in the body of the called subroutine itself, as it is done in the PDP-8.

Probably the *classic* subroutine calling method uses a JSR 300 in location 200 to cause the value 201 to be stored in location 300 and execution to start at location 301. This is the method of the PDP-8, CIP/2200 and many other minicomputers. This, as was suggested earlier, is not a very nice subroutine mechanism because it prevents you from placing the subroutine code in ROM and PROM. If you really must use ROM on a PDP-8 there are ways to do it but they are not nice. I will leave the discovery of such a method as an exercise for the reader. Imagine you have a 6800 or 8080 but that you must not use the subroutine CALL or JSR. Can you then devise a method for calling ROM subroutines? It can be done, and that is what you must do on the PDP-8.

Anyway, the main improvement in the instruction set of the Nova over the Z-80's set is the larger number of addressing modes and the ability to use any memory location for a counter or vector. You will miss the ability to jump anywhere with one instruction, but you will quickly get used to the need for a nearby vector for distant references. A few memory locations on the Nova are quite special when used as indirect vectors. The 8 locations from 10 to 17(hex) are automatically incremented when used as an indirect vector while the 8 locations from 18 to 1F(hex) are automatically decremented by such use. Only the Nova does not (normally) speak hex. The incrementing locations are octal 020 to 027 and the decrementing locations are octal 030 to 037.

Are you ready to run out and buy that Fairchild Nova chip when Fairchild offers it for $10 or *more*? The extra addressing modes are really great but the conditional instructions are liable to drive you right up the wall and maybe completely over the wall. The Nova does not have conditional jumps, calls or returns. The only conditional instructions are *skips,* which skip the very next instruction upon a *true* result. Frequently the next instruction is a jump but it need not be so. The Nova does not have the several typical flag bits for zero, parity, etc., that you are accustomed to, but the Nova *does* have a carry bit that can be set, cleared, etc., and which can be tested either alone or in combination with a register (accumulator).

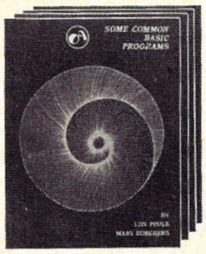
Actually, the conditional tests on the Nova are not all that bad after you grow accustomed to them, but their use is not as straightforward as it is on most 8-bit microcomputer chips. The conditional tests and much else is discussed nicely in a very good book, *How to Use the NOVA Computers* by William English and published by Data General Corp., Southboro, Massachusetts. ■

OSBORNE & ASSOCIATES, INC.

The World Leaders In Microprocessor Books

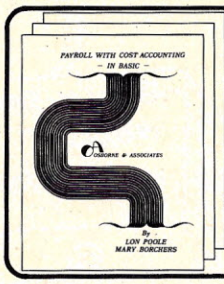
Many books on microprocessors and their use are now on the market, and most of them have names that sound alike. But Osborne & Associates' books have dominated this market since 1975, when our first book appeared. With rave reviews from all over the world, with more than three hundred university text adoptions, our books are all best sellers. In fact, "An Introduction To Microcomputers: Volume I — Basic Concepts" now holds the world's record in sales volume for any computer textbook sold for a profit.

If you want information on microprocessors, begin with the Osborne books.



"Some Common BASIC Programs". By Lon Poole and Mary Borchers.

76 short practical programs with descriptions and examples. Most can be used exactly as listed on any microcomputer with any version of BASIC. 200 pages.
Book No.: 6001 $7.50



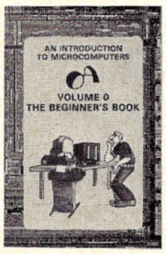
BASIC Business Programs

Each book has complete program listings in BASIC, user's manual, and thorough documentation. By Lon Poole and Mary Borchers.

"Payroll With Cost Accounting — In BASIC. 400 pages. Book No.: 22002 $12.50

"Accounts Payable And Accounts Receivable. Book No.: 23002 $12.50 Available early 1978.

"General Ledger" Book No.: 24002. $12.50 Available mid 1978.

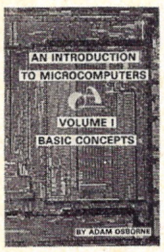


"An Introduction To Microcomputers: Volume 0 — The Beginner's Book" By Adam Osborne.

This is the book for the absolute beginner. Assuming that you know nothing about computers, math or science of any kind, this book explains what computers are all about — and it takes you to the point where you can read Volume I. 300 pages.
Book No: 6001 $7.50

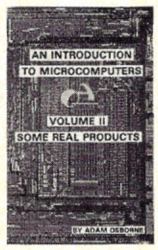
"8080A/8085 Assembly Language Programming" By Lance Leventhal.

This book is for the assembly language programmer or student; it explains assembly language programming for the 8080A and 8085 microcomputers. The book contains numerous examples. 400 pages.
Book No.: 31003 $7.50



"An Introduction To Microcomputers: Volume I — Basic Concepts" By Adam Osborne.

The world's best selling computer textbook. This book explains, clearly, concepts common to all microcomputers, yet specific to none. 350 pages.
Book No.: 2001 $7.50



"An Introduction To Microcomputers: Volume II — Some Real Products" (Revised June 1977) By Adam Osborne, Susanna Jacobson and Jerry Kane.

This book describes every common microprocessor and all of their support devices. Information is new and clearly written. Only data sheets are copied from manufacturers. 1200 pages.
Book No.: 3001 $15.00

OSBORNE & ASSOCIATES, INC. • P.O. Box 2036 • Berkeley, California 94702 • DEPT. L

		PRICE	QTY	AMT
6001	Volume 0 — The Beginner's Book	$ 7.50		
2001	Volume I — Basic Concepts	7.50		
3001A	Volume II — Some Real Products	15.00		
4001	8080 Programming For Logic Design	7.50		
5001	6800 Programming For Logic Design	7.50		
7001	Z80 Programming For Logic Design	7.50		
31003	8080A/8085 Assembly Language Programming	7.50		
21002	Some Common BASIC Programs	7.50		
22002	Payroll With Cost Accounting	12.50		

- 6-1/2%, SF Bay Area residents only
- 6%, California residents outside SF Bay Area
- Payment must be enclosed for orders of 10 books or less.
- I have enclosed: ☐ check ☐ money order

TOTAL ____
Sales Tax (Calif. residents only) ____
Shipping Charges ____
TOTAL AMOUNT OF PURCHASE ____

NAME ____
ADDRESS ____
CITY ____ STATE ____ ZIP ____ PHONE ____

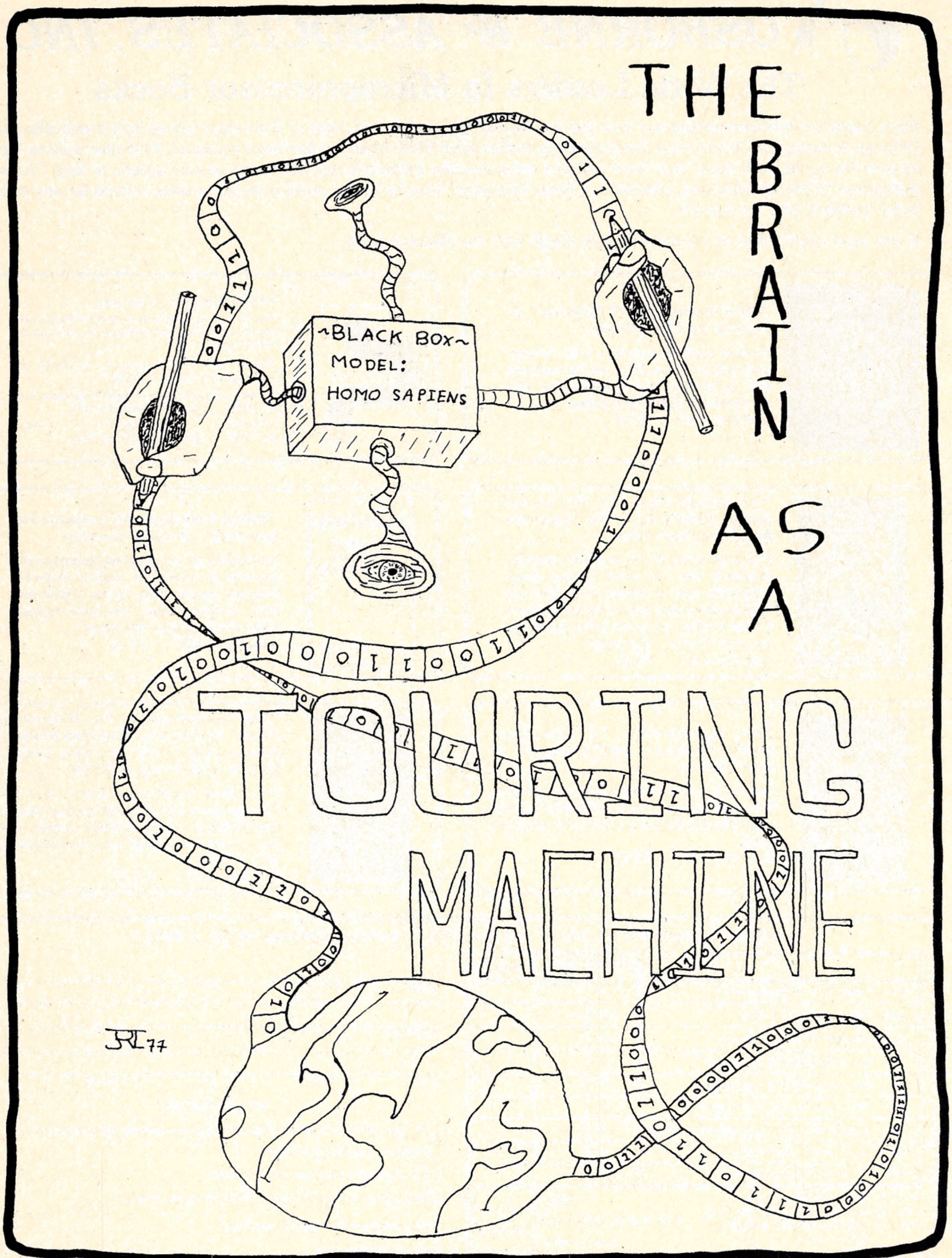
SHIPPING CHARGES

☐ 4th class (no charge, allow 3-4 weeks within USA, not applicable to discounted orders)
☐ $.50 per book, UPS (allow 10 days)
☐ $1.50 per book, special rush shipment by air
☐ $3.00 per book, foreign air mail. No foreign surface mail orders.

Shipping charges for bulk orders to be arranged.

Please send information on:
☐ consignments ☐ dealer discounts ☐ foreign distributors

F2


THE BRAIN AS A TOURING MACHINE
~BLACK BOX~
MODEL:
HOMO SAPIENS
JRL 77

Topics in Logic

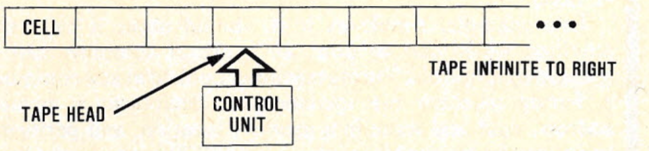
Turing Machines

John Lees

In what may be the last in this series of articles, for a while at least, we are going to climb into our Turing machine and drive off into the sunrise of automata theory. We aren't going to go very far, since automata theory becomes very complicated rather quickly, but we'll explore the basic concept. A. M. Turing, in case you're wondering, was an English mathematician who did pioneering work in computer science back in the beginning and even before. Perhaps most well known for the proposed test of machine intelligence which bears his name, Turing died in 1954.

The Turing machine is a logical, not a mechanical, device, although you can certainly build or simulate one. As such it is possible to present the concept in a number of ways, so don't be confused if you see Turing machines discussed elsewhere using a different notation. A Turing machine is a type of recognizing device, generally accepted as defining what computer scientists call a *procedure*. It can be shown that a Turing machine can be used to describe any computation which any digital computer can perform.

What is a procedure? You probably have a loose idea of a procedure as a sequence of steps performed to achieve some desired result, and that is essentially correct. A procedure is a finite sequence of steps performed to achieve a desired result but not guaranteed to achieve the result. A sequence of steps which always achieves a result is an *algorithm*. An algorithm always *halts;* a procedure may or may not halt. This distinction is obviously of great importance to the writer of a program. An infinite loop is a procedure.



From the diagram of our elementary Turing machine you can immediately see that it is very simple. Yet any program run on a digital computer can be reduced to the point where it can be represented by just such simplicity. Next time a computer tries to intimidate you, remind it that it's just a Turing machine putting on airs.

Our elementary Turing machine has a control unit, essentially a CPU, and a tape head with which it can read or write one symbol at a time in one cell of the tape at a time. The tape can be moved left or right one cell at a time, infinitely to the right. Initially a number of cells on the left contain symbols; the remainder of the tape is blank. The tape head starts out on the leftmost cell.

So what does a Turing machine do when you turn it loose? It makes a sequence of moves and in each move three things happen:

1. The machine changes state.
2. A symbol is written in the current cell, replacing the symbol or blank previously there.
3. The tape head is moved left or right one cell.

Each of these three things always happens each move, although the machine may change to the same state and may write back the same symbol it read.

The set of *states* amount to a program. Each state specification has the following form:

(CURRENT STATE, SYMBOL) → (NEW STATE, NEW SYMBOL, DIRECTION)

This amounts to a programming language with only one statement type, which means: If in the CURRENT STATE and if SYMBOL is under the tape head (SYMBOL is the contents of the current cell), then change state to NEW STATE, write NEW SYMBOL to the current cell and move one cell in DIRECTION. Then the machine evaluates the new cell in light of the new state, etc.

One state will be indicated as the *start state* or initial state and one or more as the *final state*. When a final state is reached, the Turing machine is said to have *recognized* the input on the tape and halts. Usually the initial symbols on the tape are considered to be a string in some formal language and the Turing machine is said to recognize the sentence when it halts by going into a final state.

If the Turing machine does not recognize the sentence, then it may either reach an invalid state; that is, the wrong symbol is under the head for the current state, and halt, or it may simply run forever, never halting. When the input is considered as a sentence in a language, the states have a close relationship to the grammar for the language.

Now let's consider an example of a specific Turing machine, one which recognizes the language $\{L = 1^n0 \mid n \geqslant 1\}$. This language is a string of at least one "1" followed by exactly one "0". The states of our machine might look like this:

$(S_1,1) \rightarrow (S_2,X,R)$
$(S_2,1) \rightarrow (S_2,X,R)$
$(S_2,0) \rightarrow (S_3,Y,R)$
$(S_3,B) \rightarrow (S_4,B,L)$

The start state is S_1, the final state is S_4. "1" and "0" are symbols which may appear in cells of the tape, "B" stands for a blank cell. "X" and "Y" are symbols written on the tape so the machine can determine if that cell has already been read, although that is not necessary in our example. "R" and "L" are the direction the head is moved to the next cell.

Showing the moves of the Turing machine for a string it recognizes;

$\underline{1}110$	$(S_1,1) \rightarrow (S_2,X,R)$
$X\underline{1}10$	$(S_2,1) \rightarrow (S_2,X,R)$
$XX\underline{1}0$	$(S_2,1) \rightarrow (S_2,X,R)$
$XXX\underline{0}$	$(S_2,0) \rightarrow (S_3,Y,R)$
XXXY_	$(S_3,B) \rightarrow (S_4,B,L)$

The last move results in the machine entering its final state, S_4, and halting. So the machine recognizes the string "1110" as a valid string in the language L.

Now let's look at two examples where the same Turing machine does not recognize the input string:

1100 $(S_1,1) \rightarrow (S_2,X,R)$
X100 $(S_2,1) \rightarrow (S_2,X,R)$
XX00 $(S_2,0) \rightarrow (S_3,Y,R)$
XXX0

In state S_3, "0" is not valid as the contents of the cell under the tape head, So the machine halts in error, not recognizing the input string.

1111... $(S_1,1) \rightarrow (S_2,X,R)$
X111... $(S_2,1) \rightarrow (S_2,X,R)$
XX11... and on and on and...

With an infinite string of "1"s as input, our Turing machine will neither halt in the final state nor reach an invalid state. It will continue forever.

Actually in the formal definition of a Turing machine, we do not allow the tape to initially contain an infinite string. The point of the last example is that a Turing machine produces a useful result *only* if it halts. That it doesn't halt proves nothing — the next symbol *could* be the one that would result in the machine recognizing and halting. This is why we say that a Turing machine defines a procedure, not an algorithm.

In the process of recognizing a valid input string, the Turing machine can perform can perform computations. Consider a machine that recognizes the following strings: "0+0=", "0+1=", "1+0=", and "1+1=":

$(S_1,0) \rightarrow (S_2,Y,R)$	$(S_1,1) \rightarrow (S_3,X,R)$
$(S_2,+) \rightarrow (S_4,Z,R)$	$(S_3,+) \rightarrow (S_5,Z,R)$
$(S_4,0) \rightarrow (S_6,Y,R)$	$(S_5,0) \rightarrow (S_7,Y,R)$
$(S_4,1) \rightarrow (S_8,X,R)$	$(S_5,1) \rightarrow (S_9,X,R)$
$(S_6,=) \rightarrow (S_{10},Q,R)$	$(S_7,=) \rightarrow (S_{11},Q,R)$
$(S_8,=) \rightarrow (S_{12},Q,R)$	$(S_9,=) \rightarrow (S_{13},Q,R)$
$(S_{10},B) \rightarrow (S_{10},O,R)$	$(S_{11},B) \rightarrow (S_{11},1,R)$
$(S_{12},B) \rightarrow (S_{12},1,R)$	$(S_{13},B) \rightarrow (S_{13},2,R)$

Start state = S_1
Final States = S_{10}, S_{11}, S_{12}, S_{13}

Here's how this Turing machine processes one of the valid input strings:

1+0= $(S_1,1) \rightarrow (S_3,X,R)$
X+0= $(S_3,+) \rightarrow (S_5,Z,R)$
XZ0= $(S_5,0) \rightarrow (S_7,Y,R)$
XZY= $(S_7,=) \rightarrow (S_{11},Q,R)$
XZYQ_ $(S_{11},B) \rightarrow (S_{14},1,R)$
XZYQ1_

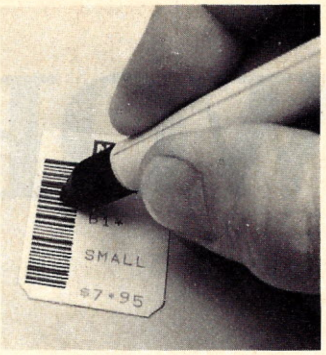
The machine has halted in one of its final states after computing that 1+0=1. A Turing machine to handle generalized arithmetic functions would have a very large number of states, but it could be done.

This example, simple as it is, should make it clear that anything which can be done on a digital computer can be restated in terms of a Turing machine. Looked at as a computer, a Turing machine can increment or decrement a memory address, read and write in a memory location, and perform a compare. A very simple machine, indeed! This proves the hard-to-believe fact that, given enough time and storage, anything you can compute on one computer can be computed on any other computer. So don't feel in awe of an IBM 370. Just buy a lot of memory and practice patience.

What we have been discussing is called an elementary Turing machine, because more complicated versions exist. The tape can be infinite in both directions, more than one tape (but still one tape head) may be used, or instead of a tape a two-dimensional grid may be used, with four possible moves from each cell. All of these modifications can be proved to be equivalent to the elementary Turing machine. Just as with computers, moving up to a more sophisticated model doesn't let you do anything fundamentally different — you are just able to do things more easily, with fewer steps.

Remembering that all current digital computers are fundamentally equivalent to a Turing machine, you might want to ponder for a while what it would mean if someone succeeded in simulating human intelligence on a computer. Is that really all we have between our ears, a Turing machine? ■

Retail Marking Code



by Susan Hastings

Years ago, manufacturers of retail marketing equipment recognized the need for automated systems of merchandise marking, but, because there was no alternative at the time, each manufacturer was forced to develop his own standard for automated data input. As equipment manufacturers and retailers began to recognize the difficulty of supply the hardware and software necessary to support all of the various systems, industry-wide marking standard was urgently sought. In 1966, after perhaps the speediest example of technological cooperation in history, OCR-A was adopted as a U.S. general purpose data processing standard for merchandisers; internationally it was adopted in 1969. But until recently, a merchandise identification system that was both technically sound and economically feasible had not been available. Now though, a chain of recent hardware and software advances have forged the missing link between the Universal Vendor Marking Standard and systems technology that will make it possible for retail automation to move into a new and more productive era.

All this progress however, will remain meaningless until it is actually implemented. In a speech to the National Retail Merchants Association's 65th annual convention, William S. Anderson, chairman and President of NCR urged everyone associated with retailing — shoppers, merchants, merchandise manufacturers, and especially, equipment manufacturers — to assist in the implementation of the UVM in order that its great potential might be proven.

Anderson said that retail equipment manufacturers can support the new Universal Standard because:

• It is a standard chosen and promoted by the general merchandise retailing industry rather than a single equipment manufacturer.

• It is human readable as well as machine readable, offering both internal systems benefits and external customer benefits.

• It is a standard which can be printed by relatively inexpensive mechanisms and thus can be implemented at relatively low cost.

• It can be used with virtually any retail medium including credit cards, invoices and other control devices.

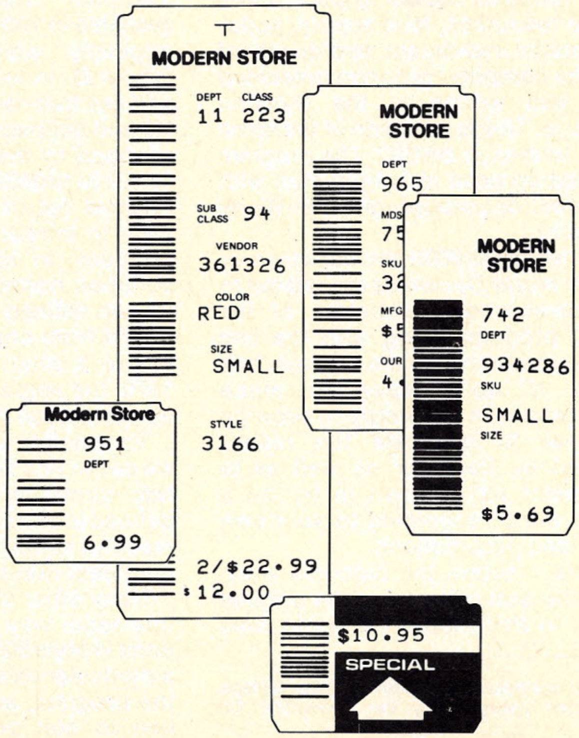
Retailers, Anderson stated, will benefit from OCR-A scanning because with it they will have a powerful new tool for achieving inventory control at the unit level rather than at the dollar level, thus, buying and selling merchandise can be accomplished more efficiently and more profitably. This benefit will be passed on to merchandise manufacturers as stock control is improved, orders are filled more efficiently, and there is more timely feedback on product reception in the marketplace.

Finally, the shopping public will find that UVM brings them new levels of shopping convenience, savings of time and greater protection against error as well as a better selection of merchandise through a reduction of out-of-stock situations.

A UVM standard makes computerized scanning systems inexpensive. And new technology will be constantly shaving away at equipment costs.

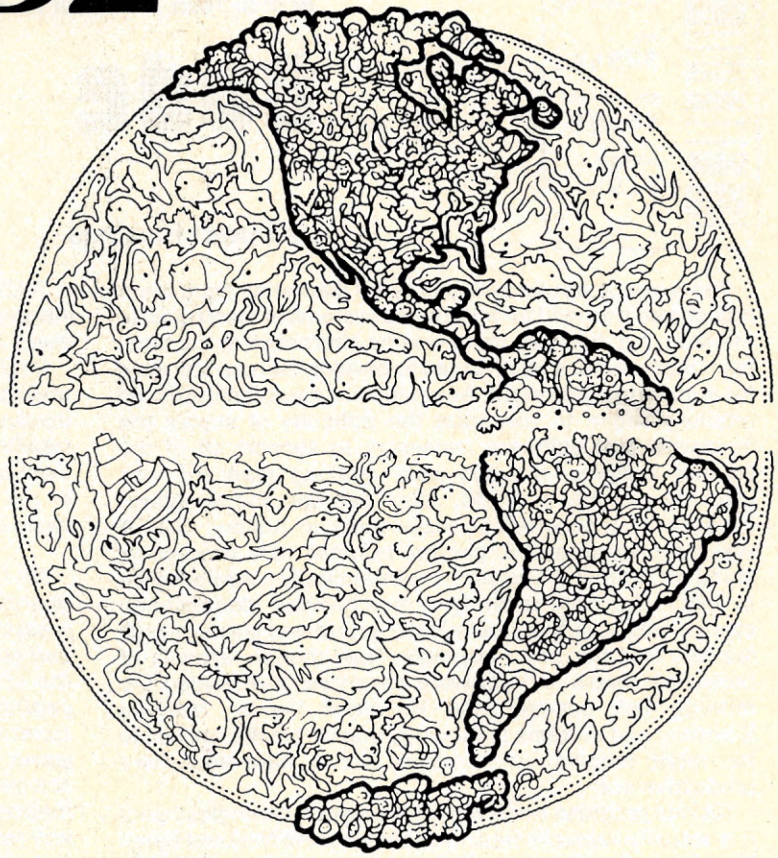
Implementing any new standard is a big and complex undertaking. This is especially true when the standard involves an industry with the tremendous diversity of retailing and its many thousands of suppliers. But if everyone in every facet of retailing is aware of the benefits of the new standard, perhaps the UVM will become the basis of the Retail Revolution of 1976.

Or 1977, or 1978, or 1979, etc. As a result of a variety of unanticipated ills (malfunctioning computers, union pressure, actions by the Consumer Federation of America and others, shoppers objections to price markings only on the shelves, etc.), UVM (or UPC) has had a rough time getting off the ground. The fact that some of the major supermarket chains have decided to play a wait-and-see game hasn't helped either. Will UVM and Supermarket scanning be a repeat performance of the great Eastern Railroads Debacle or the Alaska Pipeline Botch? Only time will tell, but in the meantime the name of technology gets dragged through a few more feet of mud and if it eventually emerges victorious, is it really a victory? — DHA ■



WORLD2

Here's your chance to create a better world, by experimenting with a simulation model.



James L. Murphy*

In [1] Jay W. Forrester presented a world population model interrelating many variables including population, natural resources, food supply, pollution, capital investment, and quality of life. The computer program presented there was written in the Dynamo language. This is a version of the same model written in BASIC. The program can best be used in conjunction with the model description given in *World Dynamics*.

In lines 5000-6000 the variables as used in *World Dynamics* are related to the corresponding variables of the BASIC program along with the line number in which they are defined. This makes it quite easy to make modifications in the program, allowing the user to duplicate the results reported by Forrester as well as to experiment on his own to try for a "better world" according to your own values and assumptions.

Figure 1 shows the outcome when using the assumptions of the original model. In this model the world runs down for lack of natural resources. You can test what would happen if you assumed the natural resources available in 1900 were double what was originally assumed available (see Figure 2). In line 5460 of the program, you see that natural resources initial is N1 and defined in line 2270 which was changed in the run of Figure 2 from 900E9 to 1800E9. This caused pollution to grow out of control, so one might want to introduce a pollution-control program. In line 5610, we see that pollution normal (after switch time) is S4 and defined in line 2400. By changing S4 from one to one-half, we see in Figure 3 what happens if, starting in 1970, the amount of pollution produced per person is somehow cut in half.

To simulate a birth-control program, we see in line 5050 that B2 is the birth-rate normal (after switch time) and is defined in line 2110. Figure 4 shows the result of setting the birth rate equal to the death rate beginning in 1970.

Many other changes can be made in this same way by using the descriptions in lines 5000-6000 to locate where a particular variable is to be defined in the program, and then modifying that line to suit your own assumptions regarding that particular variable.

There are three different sets of five variables each whose graphs may be produced. The variable A9 in line 1680 determines which of the three sets of variables is to be graphed (see Figure 5). If you would rather have a list of numbers for the variables rather than see their graphs, simply change line 2500 as in Figure 6.

There are many assumptions made in *World Dynamics* as to the relationship between various variables within the model. You may change these assumptions by changing the corresponding DATA statements in lines 1240 to 1670. For example, to change the way in which you believe the death rate is affected by pollution, you would alter the values in line 1510. It is fairly easy to do this if you first look at page 41 of [1] where the assumptions made by Forrester regarding these two variables are given. With these few examples and the program listing, I think you will now be able to create your own future worlds.

*Dept. of Mathematics, California State College, 5500 State College Pkwy, San Bernardino, CA 92407.

[1] Forrester, Jay W. *World Dynamics*. Wright-Allen Press, Inc., Cambridge, Massachusetts, 1971.

```
1000 !*******************************************************************
1010 !
1020 !A PROGRAM WRITTEN BY JAMES L. MURPHY
1030 !AT CALIF. STATE COLLEGE SAN BERNARDINO
1040 !BASED ON A MODEL BY JAY W. FORRESTER
1050 !IN "WORLD DYNAMICS", WRIGHT-ALLEN PRESS 1971.
1060 !
1070 !*******************************************************************
1080 !
1090 !A DESCRIPTION OF VARIABLES BEGINS AT LINE 5000
1100 !
1110 DIM Y(21), D(21,10)
1120 DIM P(51), C(5)
1140 PRINT "WORLD POPULATION MODEL 9-15-77"
1160 MAT D = ZER
1170 MAT Y = ZER
1180 FOR I = 0 TO 21
1190 FOR J = 0 TO 10
1200 READ D(I,J)
1210 NEXT J
1220 NEXT I
1230 RESTORE
1240 REM BRCM FROM CR
1250 DATA 1.05, 1, .9,.7,.6,.55,0,0,0,0,0
1260 REM DRCM FROM CR
1270 DATA .9,1,1.2,1.5,1.9,3,0,0,0,0,0
1280 REM FCM FROM CR
1290 DATA 2.4,1,.6,.4,.3,.2,0,0,0,0,0
1300 REM QLC FROM CR
1310 DATA 2,1.3,1,.75,.55,.45,.38,.3,.25,.22,.2
1320 REM FPCI FROM CIRA
1330 DATA .5,1,1.4,1.7,1.9,2.05,2.2,0,0,0,0
1340 REM POLCM FROM CIR
1350 DATA .05,1,3,5.4,7.4,8,0,0,0,0,0
1360 REM NREM FROM NRFR
1370 DATA 0,.15,.5,.85,1,0,0,0,0,0,0
1380 REM QLM FROM MSL
1390 DATA .2,1,1.7,2.3,2.7,2.9,0,0,0,0,0
1400 REM NRMM FROM MSL
1410 DATA 0,1,1.8,2.4,2.9,3.3,3.6,3.8,3.9,3.95,4
1420 REM CIM FROM MSL
1430 DATA .1,1,1.8,2.4,2.8,3,0,0,0,0,0
1440 REM DRMM FROM MSL
1450 DATA 3,1.8,1,.8,.7,.6,.53,.5,.5,.5,.5
1460 REM BRMM FROM MSL
1470 DATA 1.2,1,.85,.75,.7,.7,0,0,0,0,0
1480 REM FPM FROM POLR
1490 DATA 1.02,.9,.65,.35,.2,.1,.05,0,0,0,0
1500 REM DRPM FROM POLR
1510 DATA .92,1.3,2,3.2,4.8,6.8,9.2,0,0,0,0
1520 REM BRPM FROM POLR
1530 DATA 1.02,.9,.7,.4,.25,.15,.1,0,0,0,0
1540 REM QLP FROM POLR
1550 DATA 1.04,.85,.6,.3,.15,.05,.02,0,0,0,0
1560 REM BRFM FROM FR
1570 DATA 0,1,1.6,1.9,2,0,0,0,0,0,0
1580 REM DRFM FROM FR
1590 DATA 30,3, 2,1.4,1,.7,.6,.5,.5,0,0
1600 REM CFIFR FROM FR
1610 DATA 1,.6,.3,.15,.1,0,0,0,0,0,0
1620 REM QLF FROM FR
1630 DATA 0,1,1.8,2.4,2.7,0,0,0,0,0,0
1640 REM CIQR FROM QLM/QLF
1650 DATA .7,.8,1,1.5,2,0,0,0,0,0,0
1660 REM POLAT FROM POLR
1670 DATA .6,2.5,5,8,11.5,15.5,20,0,0,0,0
1680 A9 = 1
1690 C$ = "PNSQC"
1700 IF A9 = 1 THEN 1740
1710 C$ = "MRIWA"
1720 IF A9 = 2 THEN 1740
1730 C$ = "GFLBX"
1740 CHANGE C$ TO C
1750 B$(1) = "P2 - POPULATION"
1760 B$(2) = "N2 - NATURAL RESOURCES"
1770 B$(3) = "S7 - POLLUTION RATIO"
1780 B$(4) = "Q2 - QUALITY OF LIFE"
1790 B$(5) = "C2 - CAPITAL INVESTMENT"
1800 IF A9 = 1 THEN 1930
1810 B$(1) = "F8-MATERIAL STAND.OF LIV."
1820 B$(2) = "N6-NAT.RESOURCE USAGE"
1830 B$(3) = "G4-CAP.INVEST.RATIO"
1840 B$(4) = "G5-CAP.INVEST.RAT.AGRIC."
1850 B$(5) = "C4-CAP.INVEST.AG.FRAC."
1860 IF A9 = 2 THEN 1990
1870 B$(1) = "S6-POLLUTION GENERATED"
1880 B$(2) = "F6-FOOD RATIO"
1890 B$(3) = "P2/R3-LIFE EXPECTANCY"
1900 B$(4) = "B3/P2-BIRTH RATE"
1910 B$(5) = "G3-CROWDING RATIO"
1920 GO TO 2050
1930 W(1) = 8E9
1940 W(2) = 2E12
1950 W(3) = 40
1960 W(4) = 2
1970 W(5) = 2E10
1980 IF A9 = 1 THEN 2100
1990 W(1) = 4
2000 W(2) = 20E9
2010 W(3) = 4
2020 W(4) = 4
2030 W(5) = 1
2040 IF A9 = 2 THEN 2100
2050 W(1) = 40E9
2060 W(2) = 2
2070 W(3) = 100
2080 W(4) = .1
2090 W(5) = 2
2100 B1 = .026
2110 B2 = .026
2120 C1 = .4E9
2130 C2 = C1
2140 C3 = .2
2150 C4 = C3
2160 C5 = 15
2170 C6 = .3
2180 C7 = .025
2190 C8 = .025
2200 F1 = 1
2210 F3 = 1
2220 F4 = 1
2230 F5 = 1
2240 F7 = 135E6
2250 G1 = .05
2260 G2 = .05
2270 N1 = 900E9
2280 N2 = N1
2290 N3 = 1
2300 N4 = 1
2310 P1 = 1.65E9
2320 P2 = P1
2330 P3 = 26.5
2340 Q1 = 1
2350 R1 = .014
2360 R2 = .014
2370 S1 = .2E9
2380 S2 = S1
2390 S3 = 1
2400 S4 = 1
2410 S5 = 3.6E9
2420 T1 = 1900
2430 T2 = 2100
2440 T3 = 1970
2450 T4 = 1
2460 T5 = T1
2470 L3 = 5
2480 M3 = 5
2490 M2 = -1
2500 W1 = T2 + 1
2510 W2 = 20
2520 PRINT
2720 IF W1 < T2 THEN 2780
2730 FOR I = 1 TO 5
2740PRINT MID(C$,I,1);" REPRESENTS ";B$(I);"--- SCALED TO";W(I)
2750 NEXT I
2760 PRINT
2770P$="         0.2        0.4        0.6        0.8        1.0 "
2775PRINT"       0.0";P$
2780 IF T5<T3 THEN 2860
2790 B1 = B2
2800 R1 = R2
2810 F4 = F5
2820 S3 = S4
2830 C7 = C8
2840 G1 = G2
2850 N3 = N4
2860 G3 = P2/(F7*P3)
2870 I = -1
2880 X = G3
2890 K = 5
2900 GOSUB 3860
2910 GOSUB 3860
```

Grain vs. People

An article in *The Wall Street Journal* (January 20, 1975) reports that world food production is growing. It is. USDA supplies the following figures on grain production:

1970 . . . 1.11 billion metric tons
1974 . . . 1.12 " " "

Thus, grain production in 1974 was 10 million metric tons higher than in 1970.

BUT, in the same 5 year period, the world produced 355.1 million additional people, whose grain requirements each year (at the world per capita average of 311 kilos), are not 10 million more metric tons, but 110 million more metric tons.

Some have said the earth can hold many more people than it now has. From the standpoint of room alone this is so but do we want our voyage on earth to be as one on a cruise ship or one on a troop ship?

—Jack J. Combes, "Some Thoughts on Limits to Growth," *NAM Reports*, Aug. 13, 1973, p. 9.

```
2920 GOSUB 3860
2930 X = 2*G3
2940 K = 10
2950 GOSUB 3860
2960 G4 = C2/P2
2970 G5 = G4*C4/C6
2980 X = G5
2990 K = 6
3000 GOSUB 3860
3010 X = G4
3020 K = 5
3030 GOSUB 3860
3040 S6 = P2*S3*Y(5)
3050 N5 = N2/N1
3060 X = 4*N5
3070 K = 4
3080 GOSUB 3860
3090 F2 = G4*(1-C4)*Y(6)/(1-C6)
3100 F8 = F2 /F1
3110 X = F8
3120 K = 5
3130 GOSUB 3860
3140 K = 10
3150 GOSUB 3860
3160 K = 5
3170 GOSUB 3860
3180 X = 2*F8
3190 K = 10
3200 GOSUB 3860
3210 X = F8
3220 K = 5
3230 GOSUB 3860
3240 N6 = P2*N3*Y(8)
3250 G6 = P2*Y(9)*G1
3260 G7 = C2*C7
3270 S7 = S2/S5
3280 X = S7/10
3290 K = 6
3300 GOSUB 3860
3310 GOSUB 3860
3320 GOSUB 3860
3330 GOSUB 3860
3340 F6 = Y(4)*Y(2)*Y(12)*F4/F3
3350 X = F6
3360 K = 4
3370 GOSUB 3860
3380 X = 4*F6
3390 K = 8
3400 GOSUB 3860
3410 R3 = P2*R1*Y(10)*Y(13)*Y(17)*Y(1)
3420 B3 = P2*B1*Y(16)*Y(11)*Y(0)*Y(14)
3430 X = 2*F6
3440 K=4
3450 GOSUB 3860
3460 X = F6
3470 GOSUB 3860
3480 X = 2*Y(7)/Y(19)
3490 GOSUB 3860
3500 X = S7/10
3510 K = 6
3520 GO SUB 3860
3530 S8 = S2/Y(21)
3540 Q2=Q1*Y(7)*Y(3)*Y(19)*Y(15)
3550 IF T5 < W1 THEN 3590
3560 IF T5 = W1 THEN 4510
3570 IF ((T5 -W1) -W2*INT((T5-W1)/W2)) = 0 THEN 4550
3580 GO TO 3600
3590 GOSUB 3980
3600 P2 = P2 + T4*(B3 - R3)
3610 N2 = N2 - T4*N6
3620 S2 = S2 + T4*(S6 - S8)
3624 IF S2 > 0 THEN 3630
3626 S2 = 0
3630 C4 = C4 + (T4/C5)*(Y(18)*Y(20)-C4)
3640 C2 = C2 + T4*(G6 - G7)
3650 T5 = T5 + T4
3660 IF T5 < (T2 + T4) THEN 2780
3670 PRINT
3680 PRINT
3690 PRINT
3700GOTO4600
3850 REM **********TABLE INTERPOLATION ROUTINE**********
3860 I = I + 1
3870 IFX<0THEN X=0
3880 J = INT(X)+1
3890 IF X<K THEN 3950
3930 Y(I) = D(I,K)
3940 GO TO 3960
3950 Y(I) = D(I,J) + (X-J)*(D(I,J)-D(I,J-1))
3960 RETURN
3970 REM **********PLOT ROUTINE**********
3980 ON A9 GOTO 3985,4040,4100
3985 V(1)=P2
3990 V(2) = N2
4000 V(3) = S7
4010 V(4) = Q2
4020 V(5) = C2
4030 GOTO4150
4040 V(1) = F8
4050 V(2) = N6
4060 V(3) = G4
4070 V(4) = G5
4080 V(5) = C4
4090 GOTO4150
4100 V(1) = S6
4110 V(2) = F6
4120 V(3) = P2/R3
4130 V(4) = B3/P2
4140 V(5) = G3
4150 L6 = 0
4160 L2 = T5 - T1
4170 L4 = L2 - INT(L2/L3)*L3
4180 IF L4 <> 0 THEN 4500
4190MATP=CON:MATP=(32)*P
4280 M2 = M2+1
4290 IF (M2 - INT(M2/M3)*M3) <> 0 THEN 4340
4300 MAT P = CON
4310 MAT P =(45)*P
4330 L6 = 1
4340 FORL1=1TO51STEP10:P(L1)=124
4345NEXT L1:FOR L1=1TO5
4350 L4 = INT(50*V(L1)/W(L1)+.5) +1
4360 IF L4 > 51 THEN L4=51
4370 IF L4 < 1 THEN L4=1
4380 P(L4) = C(L1)
4390 NEXT L1
4400 P(0) = 51
4410 CHANGE P TO P$
4460 IF L6 = 0 THEN 4490
4470 PRINT T5;" ";P$
4480 RETURN
4490 PRINT "        ";P$
4500 RETURN
4510 PRINT "DATE","POP","NR","CI","CIAF",
4520 PRINT "POLR", "QL","CR","CIR","CIRA",
4530 PRINT "MSL","FR","NRFR","BR/P","LIFE EXP"
4540 PRINT
4550 PRINT T5,P2,N2,C2,C4,
4560 PRINT S7,Q2,G3,G4,G5,
4570 PRINT F8,F6,N5,B3/P2,P2/R3
4580 PRINT
4590 GO TO 3600
4600END

READY
```

> *"Stated most simply, if mankind is to live in the state of material well-being that technology can make possible, then, given the finite size and resources of the planet, there are just too many of us already."*
>
> —PHILIP HANDLER, *President,* National Academy of Sciences. *Bioscience,* July 1975, p. 425.

```
5000 '3420 B3    BR BIRTH RATE (PEOPLE/YEAR)
5010 '2900 Y(0)  BRCM BIRTH-RATE-FROM-CROWDING MULTIPLIER
5020 '3370 Y(16) BRFM BIRTH-RATE FROM-FOOD MULTIPLIER
5030 '3230 Y(11) BRMM BIRTH-RATE FROM-MATERIAL MULTIPLIER
5040 '2100 B1    BRN BIRTH RATE NORMAL (FRACTION/YEAR)
5050 '2110 B2    BRN1  "    "    "  (AFTER SWITCH TIME)
5060 '3320 Y(14) BRPM BIRTH-RATE-FROM-POLLUTION MULTIPLIER
5070 '3450 Y(18) CFIFR CAPITAL FRACTION INDICATED BY FOOD RATIO
5080 '3640 C2    CI CAPITAL INVESTMENT
5090 '3630 C4    CIAF CAPITAL-INVESTMENT-IN-AGRICULTURE FRACTION
5100 '2140 C3    CIAFI  "     "     "     "     "  INITIAL
5110 '2170 C6    CIAFN  "     "     "     "     "  NORMAL
5120 '2160 C5    CIAFT  "     "     "     "      "ADJUST TIME
5130 '3260 G7    CID CAPITAL INVESTMENT DISCARD
5140 '2180 C7    CIDN  "     "       "  NORMAL
5150 '2190 C8    CIDN1 "     "       "  "  (AFTER SWITCH TIME)
5160 '3250 G6    CIG CAPITAL INVESTMENT GENERATION
5170 '2250 G1    CIGN  "     "       "  NORMAL
5180 '2260 G2    CIGN1 "     "       "  "  (AFTER SWITCH TIME)
5190 '2120 C1    CII CAPITAL INVESTMENT INITIAL
5200 '3170 Y(9)  CIM CAPITAL INVESTMENT MULTIPLIER
```

```
5210 '3490 Y(20) CIQR  "        "   FROM QUALITY RATIO
5220 '2960 G4    CIR CAPITAL INVESTMENT RATIO (CAPITAL UNITS/PERSON)
5230 '2970 G5    CIRA  "        "          " IN AGRICULTURE
5240 '2860 G3    CR CROWDING RATIO
5250 '3410 R3    DR DEATH RATE (PEOPLE/YEAR)
5260 '2910 Y(1)  DRCM DEATH-RATE-FROM-CROWDING MULTIPLIER
5270 '3400 Y(17) DRFM DEATH-RATE-FROM-FOOD MULTIPLIER
5280 '3200 Y(10) DRMM DEATH-RATE-FROM-MATERIAL MULTIPLIER
5290 '2350 R1    DRN DEATH RATE NORMAL (FRACTION/YEAR)
5300 '2360 R2    DRN1  "    "     " (AFTER SWITCH TIME)
5310 '3310 Y(13) DRPM DEATH-RATE-FROM-POLLUTION MULTIPLIER
5320 '3090 F2    ECIR EFFECTIVE-CAPITAL-INVESTMENT RATIO (CAP/PERSON)
5330 '2200 F1    ECIRN     "            "          "       " NORMAL
5340 '2220 F4    FC FOOD COEFFICIENT
5350 '2230 F5    FC1 FOOD COEFFICIENT  (AFTER SWITCH TIME)
5360 '2920 Y(2)  FCM FOOD FROM CROWDING MULTIPLIER
5370 '2210 F3    FN FOOD NORMAL (FOOD UNITS/PERSON/YEAR)
5380 '3000 Y(4)  FPCI FOOD POTENTIAL FROM CAPITAL INVESTMENT
5390 '3300 Y(12) FPM FOOD FROM POLLUTION MULTIPLIER
5400 '3340 F6    FR FOOD RATIO
5410 '2240 F7    LA LAND AREA  (SQUARE KILOMETERS)
5420 '3100 F8    MSL MATERIAL STANDARD OF LIVING
5430 '3610 N2    NR NATURAL RESOURCES
5440 '3080 Y(6)  NREM NATURAL RESOURCES EXTRACTION MULTIPLIER
5450 '3050 N5    NRFR    "        "    FRACTION REMAINING
5460 '2270 N1    NRI     "        "    INITIAL
5470 '3150 Y(8)  NRMM NATURAL RESOURCES FROM MATERIAL MULTIPLIER
5480 '2290 N3    NRUN    "        "    USAGE NORMAL (UNITS/PERSON/YEAR)
5490 '2300 N4    NRUN1  "         "      "    " (AFTER SWITCH TIME)
5500 '3240 N6    NRUR NATURAL RESOURCES USAGE RATE (UNITS/YEAR)
5510 '3600 P2    P POPULATION  (PEOPLE)
5520 '2330 P3    PDN POPULATION DENSITY NORMAL (PEOPLE/SQ.KILOMETER)
5530 '2310 P1    PI POPULATION INITIAL
5540 '3620 S2    POL POLLUTION  (POLLUTION UNITS)
5550 '3530 S8    POLA POLLUTION ABSORPTION (POL.UNITS/YEAR)
5560 '3520 Y(21) POLAT    "         "    TIME  (YEARS)
5570 '3030 Y(5)  POLCM POLLUTION FROM CAPITAL MULTIPLIER
5580 '3040 S6    POLG POLLUTION GENERATION (UNITS/YEAR)
5590 '2370 S1    POLI     "    INITIAL
5600 '2390 S3    POLN POLLUTION NORMAL
5610 '2400 S4    POLN1    "         "   (AFTER SWITCH TIME)
5620 '3270 S7    POLR     "    RATIO
5630 '2410 S5    POLS POLLUTION STANDARD (POL. UNITS)
5640 '3540 Q2    QL QUALITY OF LIFE  (SATISFACTION UNITS)
5650 '2950 Y(3)  QLC QUALITY OF LIFE FROM CROWDING
5660 '3470 Y(19) QLF QUALITY OF LIFE FROM FOOD
5670 '3130 Y(7)  QLM QUALITY OF LIFE FROM MATERIAL
5680 '3330 Y(15) QLP QUALITY OF LIFE FROM POLLUTION
5690 '2340 Q1    QLS QUALITY OF LIFE STANDARD
5700 '2420 T1    INITIAL TIME (IN CALENDER YEARS)
5710 '2430 T2    ENDING TIME
5720 '2440 T3    SWITCHING TIME
5730 '2450 T4    CHANGE IN TIME  (ONE YEAR)
5740 '2460 T5    TIME
5750 '3980-4140 V(I)  I-TH VARIABLE FOR PLOT
5760 '1930-2090 W(I)  I-TH VARIABLE SCALE
5770 '2470 L3    FREQUENCY OF PLOT
5780 '2480 M3    FREQUENCY OF DATES IN PLOT
5790 '1680 A9    WHICH OF THREE GRAPHS  1  2  OR  3
5800 '2500 W1    MINIMUM PRINT TIME
5810 '2510 W2    PRINT FREQUENCY
6000 END
READY.
```

```
WORLD POPULATION MODEL 9-15-77

P REPRESENTS P2 - POPULATION--- SCALED TO .8E 10
N REPRESENTS N2 - NATURAL RESOURCES--- SCALED TO .2E 13
S REPRESENTS S7 - POLLUTION RATIO--- SCALED TO 40
Q REPRESENTS Q2 - QUALITY OF LIFE--- SCALED TO 2
C REPRESENTS C2 - CAPITAL INVESTMENT--- SCALED TO .2E 11

     0.0       0.2       0.4       0.6       0.8       1.0
1900 SC--------P----Q----¦--N------¦---------¦---------¦
     SC        ¦P     Q  ¦ N       ¦         ¦         ¦
     S C       ¦P       Q¦ N       ¦         ¦         ¦
     S C       ¦ P       Q N       ¦         ¦         ¦
     S C       ¦ P       ¦QN       ¦         ¦         ¦
1925 S--C------¦--P------¦-NQ------¦---------¦---------¦
     S  C      ¦   P     ¦ N Q     ¦         ¦         ¦
     S   C     ¦    P    ¦ N  Q    ¦         ¦         ¦
     S   C     ¦     P   ¦N   Q    ¦         ¦         ¦
     S    C    ¦      P  ¦N    Q   ¦         ¦         ¦
1950 ¦S----C---¦-------P-¦N----Q---¦---------¦---------¦
     ¦S     C  ¦        P¦N    Q   ¦         ¦         ¦
     ¦S     C  ¦         NP   Q    ¦         ¦         ¦
     ¦S      C ¦         N P Q     ¦         ¦         ¦
     ¦S       C¦        N¦  PQ     ¦         ¦         ¦
1975 ¦S--------¦C-------N¦---Q-----¦---------¦---------¦
     ¦S        ¦ C      N¦  Q  P   ¦         ¦         ¦
     ¦ S       ¦  C    N ¦  Q   P  ¦         ¦         ¦
     ¦-S       ¦   C  N  ¦ Q      P¦         ¦         ¦
     ¦ S       ¦    C N  ¦Q        P         ¦         ¦
2000 ¦--S------¦-----C---Q---------¦P--------¦---------¦
     ¦  S      ¦     N C ¦         ¦  P      ¦         ¦
     ¦   S     ¦    N Q C¦         ¦   P     ¦         ¦
     ¦   S     ¦   N Q  C¦         ¦   P     ¦         ¦
     ¦    S    ¦   NQ    C         ¦    P    ¦         ¦
2025 ¦----S----¦--N-Q----¦C--------¦----P----¦---------¦
     ¦     S   ¦ N Q     ¦C        ¦    P    ¦         ¦
     ¦     S   ¦ N Q     ¦C        ¦    P    ¦         ¦
     ¦     S   ¦N Q      ¦C        ¦    P    ¦         ¦
     ¦     S   ¦N Q      ¦C        ¦   P     ¦         ¦
2050 ¦-----S---N--Q------¦C--------¦---P-----¦---------¦
     ¦     S   N Q       ¦C        ¦  P      ¦         ¦
     ¦     S  N¦ Q       ¦C        ¦  P      ¦         ¦
     ¦     S  N¦ Q       C         ¦ P       ¦         ¦
     ¦    S   N¦ Q       C         ¦P        ¦         ¦
2075 ¦----S--N-¦-Q------C¦---------¦P--------¦---------¦
     ¦   S   N ¦Q      C ¦         P         ¦         ¦
     ¦   S   N ¦Q      C ¦        P¦         ¦         ¦
     ¦   S   N ¦Q     C  ¦        P¦         ¦         ¦
     ¦  S   N  ¦Q    C   ¦       P ¦         ¦         ¦
2100 ¦--S---N--Q----C----¦------P--¦---------¦---------¦
```

Figure 1

```
2270 N1 = 1800E9 ! TO SEE WHAT HAPPENS IF WE DOUBLE NATURAL
RESOURCES AVAILABLE.IN YEAR 1900
RUNNH
WORLD POPULATION MODEL 9-15-77

P REPRESENTS P2 - POPULATION--- SCALED TO .8E 10
N REPRESENTS N2 - NATURAL RESOURCES--- SCALED TO .2E 13
S REPRESENTS S7 - POLLUTION RATIO--- SCALED TO 40
Q REPRESENTS Q2 - QUALITY OF LIFE--- SCALED TO 2
C REPRESENTS C2 - CAPITAL INVESTMENT--- SCALED TO .2E 11

     0.0       0.2       0.4       0.6       0.8       1.0
1900 SC--------P----Q----¦---------¦---------¦----N----¦
     SC        ¦P     Q  ¦         ¦         ¦    N    ¦
     S C       ¦P       Q¦         ¦         ¦    N    ¦
     S C       ¦ P       Q         ¦         ¦    N    ¦
     S C       ¦ P       ¦ Q       ¦         ¦    N    ¦
1925 S--C------¦--P------¦--Q------¦---------¦----N----¦
     S  C      ¦   P     ¦   Q     ¦         ¦   N     ¦
     S   C     ¦    P    ¦    Q    ¦         ¦   N     ¦
     S   C     ¦     P   ¦     Q   ¦         ¦   N     ¦
     S    C    ¦      P  ¦     Q   ¦         ¦   N     ¦
1950 ¦S----C---¦-------P-¦-----Q---¦---------¦--N------¦
     ¦S     C  ¦        P¦     Q   ¦         ¦  N      ¦
     ¦S      C ¦         ¦P    Q   ¦         ¦  N      ¦
     ¦S       C¦         ¦ P  Q    ¦         ¦ N       ¦
     ¦S        C         ¦  P Q    ¦         ¦ N       ¦
1975 ¦S--------¦C--------¦----Q----¦---------¦N--------¦
     ¦ S       ¦ C       ¦    QP   ¦         ¦N        ¦
     ¦ S       ¦   C     ¦   Q   P ¦         N         ¦
     ¦ S       ¦    C    ¦  Q     P¦         N         ¦
     ¦  S      ¦     C   ¦ Q       ¦P       N¦         ¦
2000 ¦--S------¦-------C-¦-Q-------¦-P-----N-¦---------¦
     ¦   S     ¦        C¦Q        ¦   P  N  ¦         ¦
     ¦    S    ¦         QC        ¦    P N  ¦         ¦
     ¦     S   ¦        Q¦  C      ¦     N   ¦         ¦
     ¦       S ¦       Q ¦   C     ¦    N P  ¦         ¦
2025 ¦--------S¦-------Q-¦-----C---¦---N---P-¦---------¦
     ¦         ¦S     Q  ¦      C  ¦  N    P ¦         ¦
     ¦         ¦   SQ    ¦       C ¦ N     P ¦         ¦
     ¦         ¦   Q    S¦        C¦N     P  ¦         ¦
     ¦         ¦Q        ¦    S    C    P    ¦         ¦
2050 ¦------Q--¦---------¦--------NPC--S-----¦---------¦
     ¦  Q      ¦         ¦P       NC         ¦       S ¦
     ¦Q        P         ¦       C ¦         ¦         S
     ¦  QP     ¦         ¦     C N ¦         ¦         S
     ¦  P      ¦         ¦  C    N ¦Q        ¦  S      ¦
2075 ¦--P------¦---------¦C-----SN-¦---------¦---------Q
     ¦   P    S¦         C       N ¦         ¦         Q
     ¦S  P     ¦       C ¦      N  ¦         ¦         Q
     ¦ S  P    ¦       C ¦      N  ¦         ¦         Q
     ¦ S   P   ¦      C  ¦      N  ¦         ¦         Q
2100 ¦--S---P--¦------C--¦-----N---¦---------¦---------Q
```

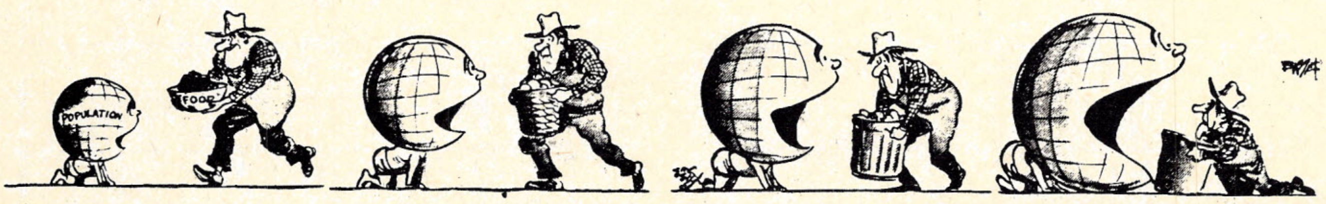
Figure 2

```
2270 N1 = 1800E9

2400 S4 = .5 ! SIMULATE POLUTION CONTROL PROGRAM BY REDUCING
POLUTION PRODCED PER PERSON TO 1/2 WHAT IT WOULD BE BEFORE 1970.

Ready

RUNNH
WORLD POPULATION MODEL 9-15-77

P REPRESENTS P2 - POPULATION--- SCALED TO .8E 10
N REPRESENTS N2 - NATURAL RESOURCES--- SCALED TO .2E 13
S REPRESENTS S7 - POLLUTION RATIO--- SCALED TO 40
Q REPRESENTS Q2 - QUALITY OF LIFE--- SCALED TO 2
C REPRESENTS C2 - CAPITAL INVESTMENT--- SCALED TO .2E 11

     0.0       0.2       0.4       0.6       0.8       1.0
1900  SC--------P----Q----|---------|---------|----N----|
      SC        |P     Q  |         |         |    N    |
      S C       |P       Q|         |         |    N    |
      S C       | P       Q         |         |    N    |
      S C       | P       | Q       |         |    N    |
1925  S--C------|--P------|--Q------|---------|----N----|
      S  C      |   P     |   Q     |         |   N     |
      S   C     |    P    |    Q    |         |   N     |
      S   C     |     P   |     Q   |         |   N     |
      S    C    |      P  |     Q   |         |   N     |
1950  |S----C---|-------P-|-----Q---|---------|--N------|
      |S     C  |        P|     Q   |         |  N      |
      |S      C |         |P    Q   |         |  N      |
      |S       C|         | P  Q    |         | N       |
      |S        C         |  P Q    |         | N       |
1975  |S--------|C--------|----Q----|---------|N--------|
      |S        | C       |    Q P  |         |N        |
      |S        |   C     |   Q   P |         N         |
      |S        |    C    |  Q      P         N         |
      |S        |      C  | Q       |P       N|         |
2000  |S--------|-------C-|Q--------|--P----N-|---------|
      |S        |         CQ        |    P N  |         |
      | S       |         QC        |     PN  |         |
      | S       |        Q|  C      |     N P |         |
      | S       |       Q |   C     |    N   P|         |
2025  |-S-------|------Q--|-----C---|---N-----P---------|
      |  S      |     Q   |      C  |  N      |P        |
      |  S      |    Q    |        C| N       | P       |
      |  S      |   Q     |         CN        |  P      |
      |  S      |   Q     |         NC        |   P     |
2050  |---S-----|--Q------|--------N|-C-------|---P-----|
      |   S     |  Q      |       N | C       |    P    |
      |   S     | Q       |      N  |  C      |    P    |
      |   S     | Q       |     N   |  C      |    P    |
      |   S     | Q       |     N   |  C      |    P    |
2075  |---S-----|-Q-------|----N----|--C------|----P----|
      |   S     |Q        |   N     |  C      |   P     |
      |   S     |Q        |  N      |  C      |   P     |
      |   S     |Q        | N       |  C      |  P      |
      |   S     |Q        | N       | C       |  P      |
2100  |---S-----|Q--------|N--------|-C-------|-P-------|
```

Figure 3

```
2110 B2 = .014  ! SIMULATE BIRTH CONTROL PROGRAM BY SETTING BASIC BIRTH
RATE EQUAL TO DEATH RATE STARTING IN SWITCH YEAR 1970.

Ready

RUNNH
WORLD POPULATION MODEL 9-15-77

P REPRESENTS P2 - POPULATION--- SCALED TO .8E 10
N REPRESENTS N2 - NATURAL RESOURCES--- SCALED TO .2E 13
S REPRESENTS S7 - POLLUTION RATIO--- SCALED TO 40
Q REPRESENTS Q2 - QUALITY OF LIFE--- SCALED TO 2
C REPRESENTS C2 - CAPITAL INVESTMENT--- SCALED TO .2E 11

     0.0       0.2       0.4       0.6       0.8       1.0
1900  SC--------P----Q----|--N------|---------|---------|
      SC        |P     Q  | N       |         |         |
      S C       |P       Q| N       |         |         |
      S C       | P       Q N       |         |         |
      S C       | P       |QN       |         |         |
1925  S--C------|--P------|-NQ------|---------|---------|
      S  C      |   P     | N Q     |         |         |
      S   C     |    P    | N  Q    |         |         |
      S   C     |     P   |N   Q    |         |         |
      S    C    |      P  |N    Q   |         |         |
1950  |S----C---|-------P-|N----Q---|---------|---------|
      |S     C  |        P|N    Q   |         |         |
      |S     C  |         NP   Q    |         |         |
      |S      C |         N P Q     |         |         |
      |S       C|        N|  PQ     |         |         |
1975  |S--------|C-------N|--P---Q--|---------|---------|
      | S       | C      N|  P     Q|         |         |
      | S       |  C    N |   P     Q         |         |
      | S       |   C  N  |   P     | Q       |         |
      |  S      |    C N  |   P     |  Q      |         |
2000  |--S------|-----C---|----P----|--Q------|---------|
      |   S     |     NC  |    P    | Q       |         |
      |    S    |    N  C |     P   |Q        |         |
      |    S    |   N    C|     P   Q         |         |
      |     S   |   N     C      P Q|         |         |
2025  |------S--|--N------|C-----Q--|---------|---------|
      |      S  | N       |C    QP  |         |         |
      |       S | N       | C Q  P  |         |         |
      |        S|N        | CQ   P  |         |         |
      |        S|N        | C    P  |         |         |
2050  |--------SN---------|QC----P--|---------|---------|
      |        SN         QC    P   |         |         |
      |        S|         QC    P   |         |         |
      |       SN|        QC    P    |         |         |
      |      SN |        QC    P    |         |         |
2075  |------SN-|-------QC|---P-----|---------|---------|
      |     S N |       C |  P      |         |         |
      |    S N  |      QC |  P      |         |         |
      |    S N  |      C  | P       |         |         |
      |   S  N  |     CQ  | P       |         |         |
2100  |---S--N--|----CQ---|P--------|---------|---------|
```

Figure 4



Gene Basset, Scripps Howard Newspapers

```
1680 A9 = 3  ! TO GRAPH A DIFFERENT SET OF FIVE VARIABLES 1,2, OR 3

READY

RUNNH
WORLD POPULATION MODEL 9-15-77

G REPRESENTS S6-POLLUTION GENERATED--- SCALED TO .4E 11
F REPRESENTS F6-FOOD RATIO--- SCALED TO 2
L REPRESENTS P2/R3-LIFE EXPECTANCY--- SCALED TO 100
B REPRESENTS B3/P2-BIRTH RATE--- SCALED TO .1
X REPRESENTS G3-CROWDING RATIO--- SCALED TO 2

     0.0       0.2       0.4       0.6       0.8       1.0
1900  |G--------|-X---B-L-|-----F---|---------|---------|
      |G        | X   B   L      F  |         |         |
      |G        | X   B   | L    F  |         |         |
      |G        |  X  B   |   L  F  |         |         |
      |G        |   X B   |     L F |         |         |
1925  |G--------|----XB---|------LF-|---------|---------|
      | G       |    BX   |       L |         |         |
      | G       |    B X  |      F L|         |         |
      | G       |    B  X |      F  |L        |         |
      |  G      |    B   X|      F  | L       |         |
1950  |--G------|---B-----X-----F---|-L-------|---------|
      |  G      |   B     | X   F   |  L      |         |
      |   G     |   B     |  X F    |   L     |         |
      |   G     |  B      |   XF    |   L     |         |
      |    G    |  B      |    FX   |    L    |         |
1975  |-----G---|--B------|----F-X--|-----L---|---------|
      |      G  |  B      |    F   X|    L    |         |
      |      G  | B       |    F    X    L    |         |
      |       G | B       |   F     | X L     |         |
      |        G| B       |   F     |  LX     |         |
2000  |---------G-B-------|---F-----|L---X----|---------|
      |         |GB       |   F     L     X   |         |
      |         | B       |   F  L  |       X |         |
      |         | BG      |   FL    |       X |         |
      |         | BG      |   L     |        X|         |
2025  |---------|-B-G-----|-L-F-----|--------X|---------|
      |         | B G     |L  F     |        X|         |
      |         | B G     L   F     |        X|         |
      |         | B  G   L|   F     |        X|         |
      |         | B  G   L|   F     |       X |         |
2050  |---------|-B--G--L-|---F-----|-------X-|---------|
      |         | B G   L |   F     |      X  |         |
      |         | B G  L  |   F     |     X   |         |
      |         |  BG  L  |   F     |     X   |         |
      |         |  B   L  |   F     |    X    |         |
2075  |---------|--B---L--|---F-----|---X-----|---------|
      |         | GB  L   |   F     |   X     |         |
      |         | GB  L   |   F     |  X      |         |
      |         |G  B L   |   F     | X       |         |
      |         G   BL    |   F     |X        |         |
2100  |---------G---BL----|---F-----|X--------|---------|
```

Figure 5

```
2500 W1 = 1900  ! TO PRINT VALUES RATHER THAN GRAPHS

READY

RUNNH
WORLD POPULATION MODEL 9-15-77

DATE           POP            NR             CI             CIAF
POLR           QL             CR             CIR            CIRA
MSL            FR             NRFR           BR/P           LIFE EXP

 1900           .165E 10       .9E 12         .4E 9          .2
 .555556E-1     .611596        .461216        .242424        .161616
 .277056        1.03861        1              .318735E-1     36.7659

 1920           .196116E 10    .88628E 12     .889143E 9     .21521
 .161558        .859049        .548192        .453377        .325238
 .503645        1.10128        .984756        .315628E-1     51.2495

 1940           .253342E 10    .859044E 12    .171298E 10    .22909
 .314741        1.01171        .708153        .676155        .516334
 .724319        1.08527        .954494        .297325E-1     61.0276

 1960           .328022E 10    .811839E 12    .298576E 10    .258357
 .579815        1.00748        .916902        .910233        .783883
 .907703        1.00869        .902044        .270773E-1     67.3233

 1980           .41307E 10     .740856E 12    .466868E 10    .295855
 1.1492         .938059        1.15463        1.13024        1.11463
 1.01631        .987254        .823174        .253664E-1     70.9236

 2000           .502579E 10    .646852E 12    .658197E 10    .321481
 2.25276        .781284        1.40483        1.30964        1.40341
 1.02345        .966456        .718724        .238568E-1     62.9535

 2020           .556544E 10    .544299E 12    .805598E 10    .331894
 3.74293        .606776        1.55568        1.4475         1.60139
 .893431        .940777        .604777        .230056E-1     47.2273

 2040           .553121E 10    .451767E 12    .858396E 10    .322477
 4.90884        .530151        1.54611        1.55191        1.66819
 .755169        .951912        .501964        .235889E-1     38.8276

 2060           .520575E 10    .377586E 12    .822329E 10    .299769
 4.77267        .490381        1.45514        1.57966        1.57844
 .612091        .969657        .41954         .024964        34.9806

 2080           .482322E 10    .322821E 12    .728486E 10    .277566
 3.52549        .452075        1.34821        1.51037        1.39743
 .471009        .975294        .358689        .264604E-1     32.6006

 2100           .43935E 10     .284492E 12    .610553E 10    .260927
 2.3075         .41795         1.22809        1.38967        1.20868
 .355871        .977047        .316103        .278062E-1     30.4718
```

Figure 6

puzzles & problems

Debugging the Raise

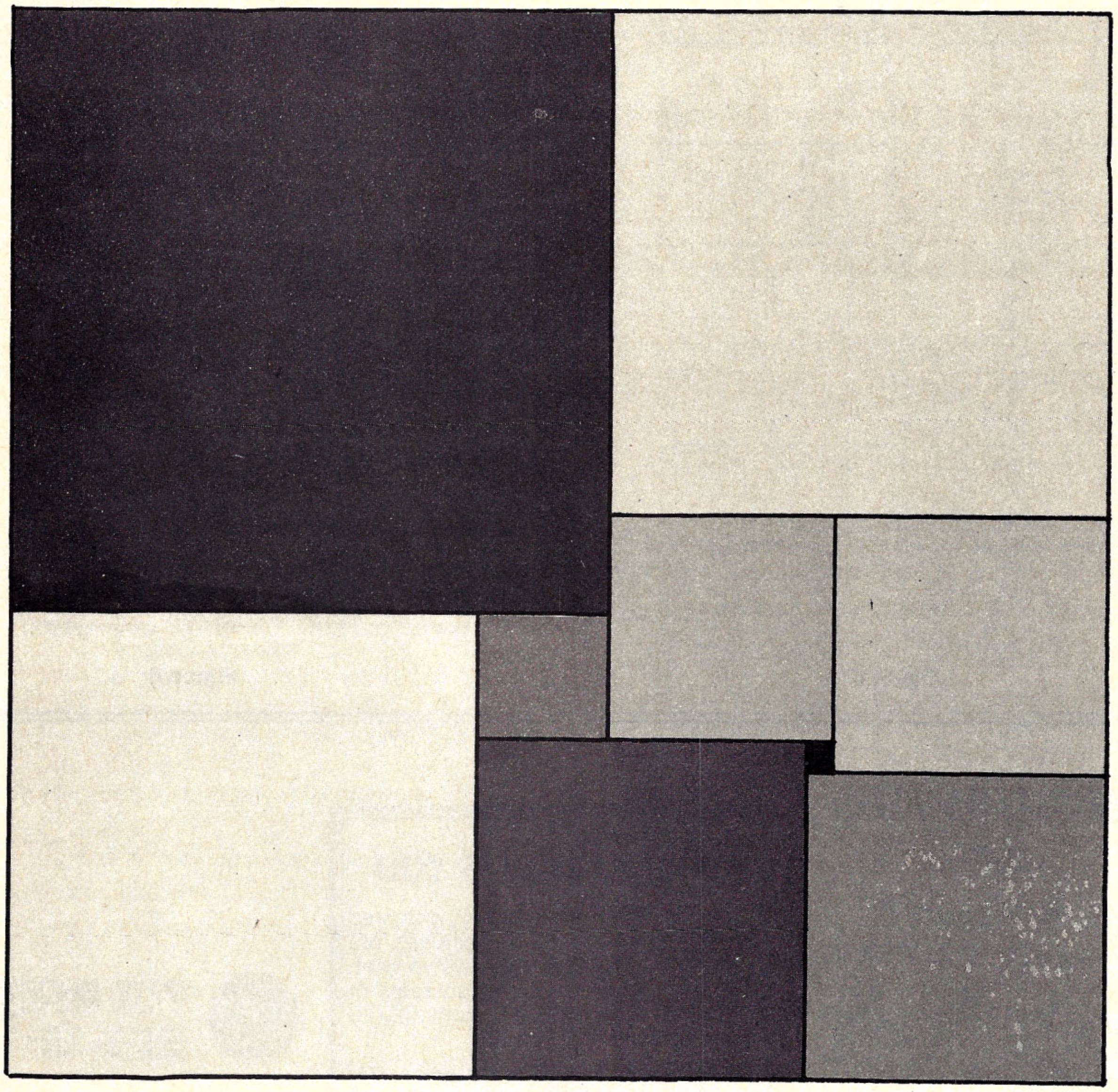
New program debuggers are paid $10,000 per year at Microbug, Inc. Every six months there is a salary review and good debuggers get a $400 annual raise. Now management has proposed that the salary review occur annually, and that good debuggers get an annual increase of $850. Good for the employees or the company? Why?

DHA

Cubiquiz

Here is a number: 94217. (a) Drop one digit and rearrange the others to produce a perfect cube. (b) Drop another and rearrange to give another perfect cube. (d) Do the same again.

Games & Puzzles



Perfect Rectangle

The reproduction is of a painting titled "Perfect Rectangle" by Mary Russell. The underlying idea is mathematical. A not-quite-square rectangle has been divided into the minimum number of squares that will fill its entire area, making it a "perfect" rectangle.

Write a program to determine for any rectangle with dimension L and W the minimum number of squares that will fill its area.

CAUTION! This rather innocent problem is not that easy to solve!

FOR ALL CUSTOMERS EXCEPT CALIF. CALL TOLL FREE 800-421-5809

MICROCOMPUTER

MICROPROCESSOR's

F-8	19.95
Z-80	25.00
Z-80A	35.00
CDP1802CD	24.95
AM2901	22.95
6502	12.95
6800	19.95
8008-1	8.75
8080A	15.95
TMS-9900TL	89.95

6800 SUPPORT

6810P	4.95
6820P	8.00
6828P	11.25
6834P	16.95
6850P	9.95
6852P	11.95
6860P	14.95
6862P	17.95
6880P	2.70

Z80 SUPPORT DEVICES

3881	12.95
3882	12.95

F-8 SUPPORT DEVICES

3851	14.95
3853	14.95

8080A SUPPORT DEVICES

8212	3.95
8214	9.95
8216	4.50
8224	4.95
8228	8.75
8238	8.00
8251	12.00
8253	28.00
8255	12.00
8257	25.00
8259	25.00

CHARACTER GENERATORS

2513 UP	6.75
2513 DOWN	6.75
2513 UP (5v)	9.95
2513 DOWN(5v)	10.95
MCM6571	10.80
MCM6571A	10.80
MCM6572	10.80
MCM6574	14.75
MCM6575	14.75

STATIC RAMS		1-24	25-99	100
21L02	(450)	1.50	1.40	1.25
21L02	(250)	1.95	1.80	1.50
21L11		4.25	4.10	3.95
1101A		1.49	1.29	1.10
2101-1		2.95	2.75	2.60
2102		1.25	1.15	1.00
2102-1		1.50	1.30	1.15
2111-1		4.00	3.50	3.25
2112-1		3.00	2.80	2.69
2114		17.95	16.95	16.50
4200A		12.95	12.50	11.95
5101C-E		11.95	11.25	10.25

WAVEFORM GENERATOR

8038	4.00
MC4024	2.50
566	1.75

KEYBOARD CHIPS

AY5-2376	13.95
AY5-3600	13.95

PROM'S

1702A	5.00
2704	15.00
2708	13.00
2716	38.00
3601	4.50
5203AQ	4.00
5204AQ	6.00
6834	16.95
6834-1	14.95
82S23B	4.00
8223B	2.70

DYNAMIC RAMS

1103	1.50
2104	4.50
2107A	3.75
2107B	4.50
2107B-4	4.00
TMS4050	4.50
TMS4060	4.50
4096	4.50
4116	42.00
MM5270	5.00
MM5280	6.00
MCM6605	6.00

UART'S

AY5-1013A	5.50
AY5-1014A	8.95
TR-1602A	5.50
TMS-6011	6.95
IM-6402	10.80
IM-6403	10.80

MISC. OTHER COMPONENTS

NH0025CN	1.70
NH0026CN	2.50
N8T20	3.50
N8T26	2.45
74367	.90
DM8098	.90
1488	1.95
1489	1.95
D-3207A	2.00
C-3404	3.95
P-3408A	5.00
P-4201	4.95
MM-5320	7.50
MM-5369	1.90
DM-8130	2.90
DM8131	2.75
DM-8831	2.50
DM-8833	2.50
DM-8835	2.50
SN74LS367	.90
SN74LS368	.90

KIM

KIM-1	245.00
6502	12.95
6520	9.00
6522	9.25
6530-002	15.95
6530-003	15.95
6530-004	15.95
6530-005	15.95

USRT

S-2350	10.95

LATE ADDITIONS

TMS 4044	$14.00
TMS 5501	$24.95

FLOPPY DISC CONTROLLER

1771B	55.95
1771B-01	59.95

8K STATIC RAM BOARD

ASSEMBLED & TESTED

250ns. $199.95
450ns. $150.00

- * WILL WORK WITH NO FRONT PANEL
- * FULL DOCUMENTATION
- * FULLY BUFFERED
- * S100 DESIGN
- * ADEQUATELY BYPASSED
- * LOW POWER SCHOTTKY SUPPORT IC'S

KIT

250ns. $169.95
450ns. $129.95

BARE BOARD W/SCHEMATIC $25.00

ADAPT YOUR MOTOROLA 6800 SYSTEM TO OUR S-100 8K RAM BOARD. KIT PRICE $12.95

JADE VIDEO INTERFACE KIT

FEATURES **$89.95**

S-100 Bus Compatible
32 or 64 Characters per line
16 lines
Graphics (128 x 48 matrix)
Parallel & Compositive video
On board low-power memory

Powerful software included for cursor, home, EOL, Scroll Graphics/Character, etc.
Upper case, lower case & Greek
Black-on-white & white-on-black

JADE PARALLEL/SERIAL INTERFACE KIT

$124.95 KIT

5 PORTS
2 PARALLEL/2 SERIAL
1 CASSETTE

EDGE CONNECTORS

S-100 Altair Spacing $4.45
S-100 Imsai Spacing $4.45

IMSAI/ALTAIR S-100 COMPATIBLE JADE Z80 KIT

—with PROVISIONS for ONBOARD 2708 and POWER ON JUMP

$135.00 EA. (2MHZ)
$149.95 EA. (4MHZ)

BARE BOARD $35.00

MOTHER BOARD

- * 13 SLOT MOTHER BOARD w/front panel slot
- * S-100 DESIGN
- * FULL GROUND PLANE ON ONE SIDE
- * RC NETWORK TERMINATION ON EVERY LINE EXCEPT PWR & GRD
- * KLUGE AREA
- * STRONG 1/8" THICK DOUBLE SIDED BOARD

BARE BOARD $35.00 KIT $85.00

THE PROM SETTER

WRITE and READ EPROM

1702A-2708-2716
5204-6834

- Plugs Directly into your ALTAIR/IMSAI Computer
- Includes Main Module Board and External EPROM Socket Unit
- The EPROM Socket Unit is connected to the Computer through a 25 Pin Connector
- Programming is accomplished by the Computer
- Just Read in the Program to be Written on the EPROM into your Processor and let the Computer do the rest
- Use Socket Unit to Read EPROM's Contents into your Computer
- Software included
- No External Power Supplies. Your Computer does it all
- Doubles as an Eight Bit Parallel I/O
- Manual included *Delivery Less than 90 days*

THE PROM SETTER
KIT $210.00 ASSEMBLED $375.00

JADE 8080A KIT

$100.00 KIT

BARE BOARD $35.00

PERSCI DISK DRIVE FOR S-100

COMPLETE
Info 2000 S-100 DISK SYSTEM
(includes dual drive, power supply, case, intelligent controller, adapter, cables, and disk monitor on EPROM) **$2,650.00**

COMPLETE TDL SOFTWARE PKG. FOR DISK **$195.00**

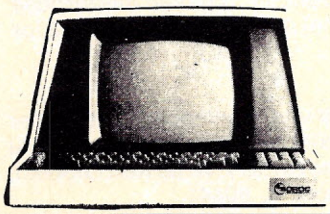
TV-1 VIDEO INTERFACE

You will want to know about the TV-1 Video to Televisior Interface Kit.
No need to buy a separate Video Monitor if you already own a TV set. Just connect the TV-1 between your system video output and the TV set antenna terminals - that's all there is to it - to convert your TV set to a Video Monitor, and at a much lower cost! **PRICE $8.95**

REAL TIME CLOCK FOR S-100 BUS

BARE BOARD $30.00 KIT $124.95

SOROC IQ 120 TERMINAL



A CAPABLE LOW COST. APPROACH TO REMOTE VIDEO DISPLAY TERMINALS.

- Cursor Control Keys Standard
- Numeric Key Pad Standard
- Line and Page Erase Standard
- Addressable Cursor Standard
- Switch Selectable Transmission from 75 to 19,200 bps Standard
- Communication Mode: HDX FDX Block
- Interfaces: Printer Interface, RS232 Extension
- RS232C Interface
- Non-Glare Read Out Screen
- Protect Mode Standard
- Tab Standard

$975 00
Assembled

Price Includes
- Block Mode
- Lower Case
- 24 Line Option
- and shipping charge is on us.

JADE

Computer Products

5351 WEST 144th STREET
LAWNDALE, CALIFORNIA 90260
(213) 679-3313

RETAIL STORE HOURS M-F 9-7 SAT. 9-5

Discounts available at OEM quantities. Add $1.25 for shipping. California residents add 6% sales tax

1978 CATALOG NOW AVAILABLE

puzzles & problems

Bionic Toads

We have been training six bionic toads to do a new trick. When placed on seven glass tumblers, as shown below, they change sides so the three black ones are to the left and the white ones to the right, with the unoccupied tumbler at the opposite end—#7. They can jump to the next tumbler (if unoccupied), or over one or two toads to an unoccupied tumbler. The jumps can be made in either direction, and a toad may jump over his own or the opposite color, or both colors. Four successive sample jumps will make everything plain: 4 to 1, 5 to 4, 3 to 5, 6 to 3. Can you show how they do it in 10 jumps? Can you write a program to solve the problem?

A man rarely succeeds at anything unless he has fun doing it.

Programming Problem . . .

SATURDAY NIGHT TENNIS SCHEDULE

Given:
- 6 tennis courts
- 6 time periods (45 minutes each)
- 18 couples

Objectives:
- each couple plays together once (1st time period)
- each person plays four times, sits out twice
- each person plays one men's (women's) doubles match once; 3 mixed doubles matches.
- each person should not play with or against the same person more than twice.

The task:
Develop a schedule that meets the objectives, and also optimizes each individual's personal schedule. (Optimum means no more than two plays back to back) It's also nice to have both members of a couple arrive and leave together — schedules permitting!

The task minus one:
What do you do if one couple doesn't show up?

Furthermore:
How about a general program for N courts, M time periods and C couples.

The Mechanical Man

Ken Lebeiko

The mechanical man is designed as a beginning exercise in flowcharting. If you're learning programming on your own, try the mechanical man flowchart; even experienced people will have fun with it. If you're teaching flowcharting or programming, follow the steps below and it will be an exciting experience for both you and your class.

1. Copy the handout on the facing page, give it to your class and explain the problem. Keep it simple; state there is a clear path to the wall. An example may help; simply put a chair about six feet from the wall in the classroom.
2. Take the first few flowcharts done (make sure they aren't perfect) and ask the students to place them on the board. This will allow the slower students to get a general idea of what is to be done.
3. Work through the flowcharts without counters first. Before starting those flowcharts with counters, draw a counter and put a number in it (87). This is to show the effect of not clearing a counter in a program. Don't make a big thing of this; just say it was left from the last program. Everyone will see the results later.
4. Now have the student whose flowchart you are working with become the mechanical man. Use a pointer and instruct the student, following every detail of his flowchart. As the student or class see errors, don't let them change the flowchart.

Ham it up; play it for all its worth. The class will have a good time and really learn a lot.

Common things to happen are: people forget to turn twice and walk off somewhere else, or they forget to turn before sitting and end on the floor, and lots more—the list is long.

5. Before the students get tired of the problem, tell them to work in groups and test each others flowcharts for the perfect one.
6. Toward the end of class ask who has the "perfect" flowchart and have him put it on the board.
7. Have a member of that group sit in the chair, but move the chair right against the wall, leaving only enough room for the student's legs.

This is called a special case. Discuss whether or not the flowchart will still work and what could be changed to make it work. This case could be used for homework. You can tell the students to get their families involved by having a parent become the mechanical man.

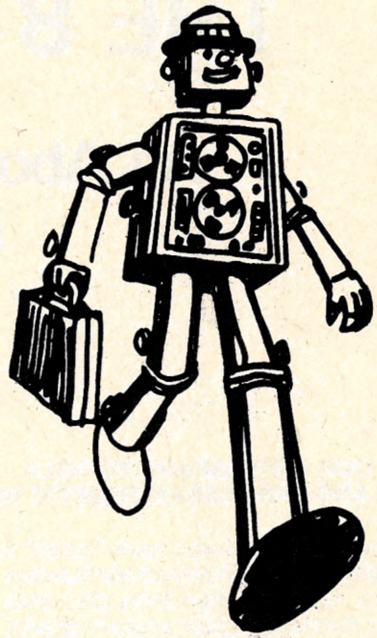
8. Another special case is placing the chair one step from the wall. Give this case as homework too. It demonstrates that it is very difficult to get an all-purpose program and that some special cases must be allowed for separately.

The author would like to invite any teacher with other ideas for teaching programming or flowcharting to share them with him (and us at *Creative* too!). Write Ken Lebeiko, Lockport Central HS, Lockport, IL 60441.

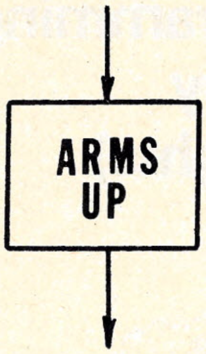
The Mechanical Man

A mechanical man is sitting on a chair, facing a wall a short distance away. Draw the flowchart of the procedure to have the mechanical man walk to the wall and return to his initial position.

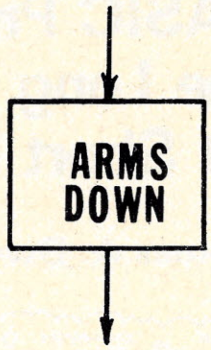
Permissible flowcharting symbols are:



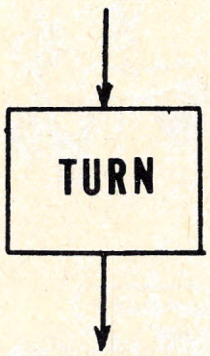
Start the procedure.



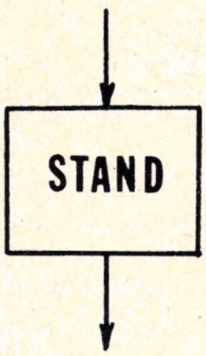
Raise arms straight out.



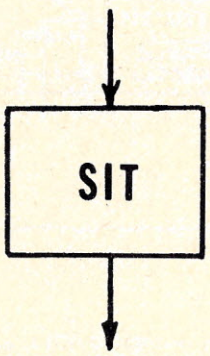
Lower arms.



Turn 90° to the right.

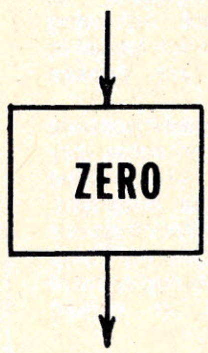


Stand up.

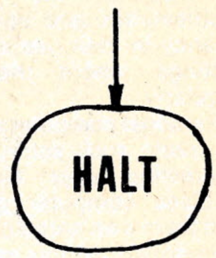


Sit down.

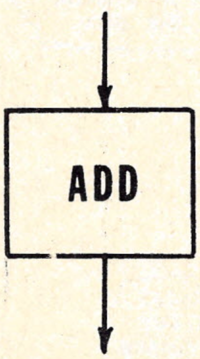
Take one step.



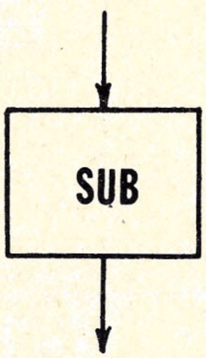
Set the counter to zero.



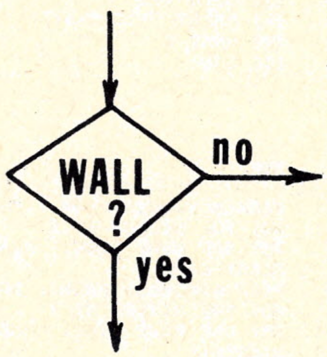
Stop the procedure.



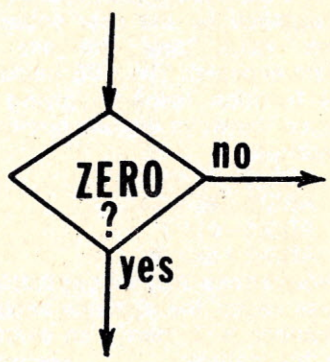
Add 1 to the counter.



Subtract 1 from the counter.



Test for touching wall. Arms must be straight forward to activate the finger-contact units.



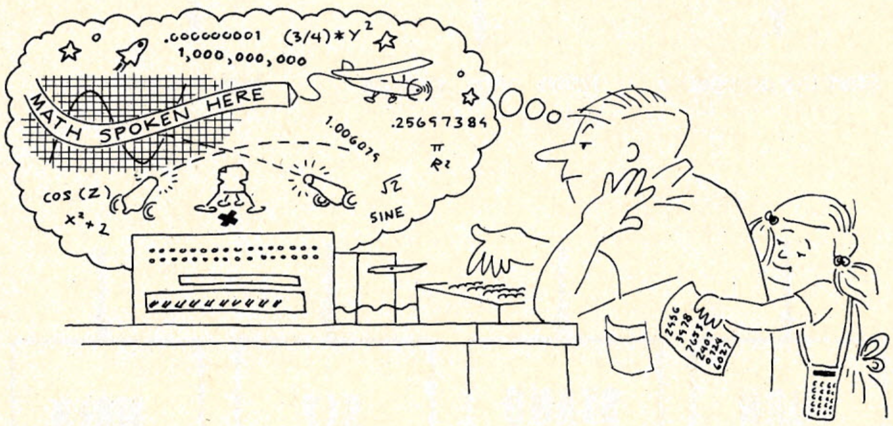
Test if counter is zero.

Part IV, the conclusion of this series, examines ABS, SQR, REM, GOSUB, and the mysterious DEF FNX.

THE 8-HOUR WONDER

All About BASIC Programming in One Long Day (or Eight Short Nights)

Thomas A. Dwyer



2.7 HOUR 7: PROGRAMS TO HELP MOM AND DAD PASS ARITHMETIC 102

Very few people who have "taken" a foreign language in school are fluent in its use. Little children from countries where that language is spoken do a lot better, and with far less fuss. The same is true of the "languages" of mathematics and science. Achieving fluency in their use is much easier in settings where they are spoken regularly.

Personal computers make it possible to create such settings in some very interesting ways. One of the best involves computer game programs, and there's an entire chapter on games coming up. In this section we'll help prepare the way by explaining some of the techniques used in writing number-oriented games.

ABS

ABS(X) is a function that "processes" X in a very simple manner. It merely changes the sign of X to +. This is useful when we want to check how close some INPUT data supplied by the user comes to another value (say, the one the program expects). The ABS (absolute value) function helps by giving the "distance" between the two numbers. For example:

ABS(8 - 5) = 3
ABS(5 - 8) = 3
ABS(5 - 2) = 3
ABS(2 - 5) = 3

As you can see, ABS tells us that in all of these cases, the distance between the number is 3. Here's an example using this feature: →

```
LIST

3 RANDOMIZE
5 FOR K=1 TO 3
10 LET R = INT(10 * RND(0) + 1)
20 PRINT "PICK A NUMBER FROM 1 TO 10";
30 INPUT N
40 IF R = N THEN 110
50 PRINT "NO, YOU MISSED BY";ABS(N - R)
60 PRINT "TRY ONE MORE TIME.  NUMBER IS";
70 INPUT N
80 IF R = N THEN 110
90 PRINT "YOU BLEW IT.  THE NUMBER WAS";R
100 GOTO 120
110 PRINT "RIGHT!!!"
120 NEXT K
130 END
```

THIS IS THE COMPUTER'S NUMBER

THIS IS YOUR NUMBER.

THIS GIVES THE AMOUNT BY WHICH N MISSED, BUT NOT THE SIGN.

```
RUN

PICK A NUMBER FROM 1 TO 10? 5
NO, YOU MISSED BY 3
TRY ONE MORE TIME.  NUMBER IS? 8
RIGHT!!!
PICK A NUMBER FROM 1 TO 10? 1
NO, YOU MISSED BY 7
TRY ONE MORE TIME.  NUMBER IS? 8
RIGHT!!!
PICK A NUMBER FROM 1 TO 10? 10
NO, YOU MISSED BY 2
TRY ONE MORE TIME.  NUMBER IS? 8
RIGHT!!!
```

WITH THESE CHOICES YOU CAN ALWAYS GET IT ON THE 2ND CHANCE.

Another use of ABS is for accepting input that is "close enough" even though not exactly the number expected. The art of getting such "ball-park" estimates is seldom taught in school, yet it's a valuable one. Here's an example of a program for practicing this:

```
LIST

3 RANDOMIZE
5 FOR K = 1 TO 10
10 LET H = INT(4000 * RND(0) + 1200)/100
20 LET M = INT(1000 * RND(0) + 500)/100
30 LET I = INT(300 * RND(0) + 300)/100
40 LET D = INT(9 * RND(0) + 1)
50 PRINT "APPROXIMATELY HOW MUCH SHOULD YOU BUDGET"
55 PRINT "FOR A TRIP OF";D;"DAYS IF--"
60 PRINT "  HOTEL COST PER DAY   = $";H
70 PRINT "  MEAL COST PER DAY    = $";M
80 PRINT "  INCIDENTALS PER DAY = $";I
90 INPUT A
100 LET C = D * (H + M + I)
110 LET E = ABS(A - C)
120 IF E/C < .10 THEN 160
130 PRINT "YOU MISSED BY $";E
140 PRINT "YOU WERE OFF BY";(E/C)*100;"%"
150 GOTO 180
160 PRINT "VERY GOOD.  YOU WERE OFF BY $";E
170 PRINT "THAT WAS AN ERROR OF ONLY";(E/C)*100;"%"
180 NEXT K
190 END


RUN

APPROXIMATELY HOW MUCH SHOULD YOU BUDGET
FOR A TRIP OF 5 DAYS IF--
  HOTEL COST PER DAY   = $ 42.84
  MEAL COST PER DAY    = $ 12.81
  INCIDENTALS PER DAY = $ 5.25
? 60
YOU MISSED BY $ 244.5
YOU WERE OFF BY 80.2956 %
APPROXIMATELY HOW MUCH SHOULD YOU BUDGET
FOR A TRIP OF 3 DAYS IF--
  HOTEL COST PER DAY   = $ 15.12
  MEAL COST PER DAY    = $ 7.03
  INCIDENTALS PER DAY = $ 4.54
? 76
VERY GOOD.  YOU WERE OFF BY $ 4.07
THAT WAS AN ERROR OF ONLY 5.08305 %
APPROXIMATELY HOW MUCH SHOULD YOU BUDGET
FOR A TRIP OF 5 DAYS IF--
  HOTEL COST PER DAY   = $ 50.22
  MEAL COST PER DAY    = $ 8.35
  INCIDENTALS PER DAY = $ 4.23
? 320
VERY GOOD.  YOU WERE OFF BY $ 6
THAT WAS AN ERROR OF ONLY 1.91083 %
APPROXIMATELY HOW MUCH SHOULD YOU BUDGET
FOR A TR^C
```

THIS SHOULD HAVE BEEN DONE MENTALLY USING 43+13+5=61 × 5=305.

Notice that we used two kinds of "error" formulas in this program. The absolute error E = ABS(A - C) gives the absolute value of the *difference* between the correct answer and the approximate answer, while the relative error E/C shows the *ratio* between this difference and the correct answer.

Why make this distinction? Well suppose you were a contractor who made a bid that missed the true cost by $1000. How serious is this? It all depends. If you take two extreme cases, you'll see why.

Case 1:

True cost	=	$50,000
Your bid	=	$49,000
Absolute error	=	$ 1,000
Relative error	=	1000/50000 = .02
Percent error	=	2%

Case 2:

True cost	=	$ 2,500
Your bid	=	$ 1,500
Absolute error	=	$ 1,000
Relative error	=	1000/2500 = .4
Percent error	=	40%

The absolute error was the same in both cases. It's the relative error that shows which one is a disaster. (Percent error also shows this, since it is merely relative error multiplied by 100.)

ABS is also handy in making sure that an input response is as requested. Here's one way this can be done:

```
LIST

10 PRINT "TYPE A POSITIVE INTEGER BETWEEN 50 AND 100."
20 INPUT A
30 IF INT(A) <> A THEN 120
40 IF ABS(75-A) > 25 THEN 140
50 PRINT "YOU HAVE OBEYED A COMPUTER."
60 PRINT "THERE IS NO HOPE."
70 STOP
120 PRINT "THAT'S NOT AN INTEGER."
130 GO TO 10
140 PRINT "OUT OF REQUESTED RANGE."
145 PRINT "READ THE INSTRUCTIONS CAREFULLY."
150 GO TO 10
160 END

RUN

TYPE A POSITIVE INTEGER BETWEEN 50 AND 100.
? 25
OUT OF REQUESTED RANGE.
READ THE INSTRUCTIONS CAREFULLY.
TYPE A POSITIVE INTEGER BETWEEN 50 AND 100.
? 7.5
THAT'S NOT AN INTEGER.
TYPE A POSITIVE INTEGER BETWEEN 50 AND 100.
? 75
YOU HAVE OBEYED A COMPUTER.
THERE IS NO HOPE.
STOP at line 70
```

If you want to be more explicit in your error messages, statement 40 can be replaced by two tests:

```
40 IF A < 50 THEN 140
45 IF A > 100 THEN 142
•
•
•
140 PRINT "TOO SMALL!"
141 GO TO 145
142 PRINT "TOO LARGE!"
•
•
•
```

SQR

We'll finish this section with a math-game program that uses the square-root function of BASIC. SQR(X) processes the number X by finding its square root and "returning" this value in the place where SQR is used. (The square root of X is a number which when multiplied by itself gives X. This means you must use positive numbers for X. Otherwise you'll get an error message.)

Example:

```
10 LET X = 25          If the number 25 is supplied
20 PRINT X, SQR (X)    to the SQR function.

RUN
25             5       The number 5 is "returned."
```

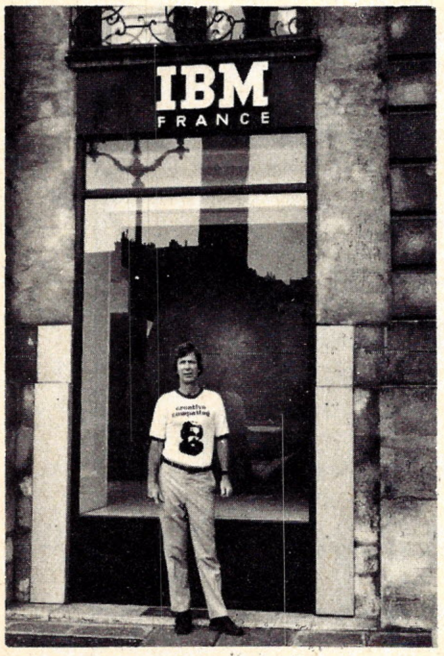
Here's a game program to practice estimating square roots:

```
LIST

5 RANDOMIZE
10 LET K = 0
20 PRINT "TO WIN THE GOLD STAR YOU NEED 3 ANSWERS IN A ROW"
25 PRINT "THAT HAVE LESS THAN 5% ERROR."
30 PRINT "-------------------------------------------------"
50 LET R = INT(100*RND(0) + 1)
60 PRINT "WHAT IS THE SQUARE ROOT OF";R
70 INPUT A
80 LET C = SQR(R)
90 IF ABS(A - C)/C < .05 THEN 130
100 PRINT "NOT TOO CLOSE.  SQUARE ROOT OF ";R;"IS";C
105 PRINT "YOU MISSED BY";100*ABS(A/C-1);"%"
106 PRINT
110 LET K = 0
120 GO TO 50
130 PRINT "NOT BAD--YOU ONLY MISSED BY";100*ABS(A/C-1);"%"
135 PRINT "SQUARE ROOT OF ";R;"IS";C
136 PRINT
140 LET K = K + 1
150 IF K < 3 THEN 50
160 PRINT "THAT'S 3 IN A ROW!        *****"
170 PRINT "  PASTE STAR HERE--       *   *"
175 PRINT "                          *****"
180 END
```

```
RUN

TO WIN THE GOLD STAR YOU NEED 3 ANSWERS IN A ROW
THAT HAVE LESS THAN 5% ERROR.
-------------------------------------------------
WHAT IS THE SQUARE ROOT OF 78
? 8.11
NOT TOO CLOSE.  SQUARE ROOT OF  78 IS 8.83176
YOU MISSED BY 8.17233 %

WHAT IS THE SQUARE ROOT OF 79
? 8.8
NOT BAD--YOU ONLY MISSED BY .992265 %
SQUARE ROOT OF  79 IS 8.88819

WHAT IS THE SQUARE ROOT OF 76
? 8.5
NOT BAD--YOU ONLY MISSED BY 2.49831 %
SQUARE ROOT OF  76 IS 8.7178

WHAT IS THE SQUARE ROOT OF 48
? 6.10
NOT TOO CLOSE.  SQUARE ROOT OF  48 IS 6.9282
YOU MISSED BY 11.9541 %

WHAT IS THE SQUARE ROOT OF 8
? 6.4
NOT TOO CLOSE.  SQUARE ROOT OF  8 IS 2.82843
YOU MISSED BY 126.274 %

WHAT IS THE SQUARE ROOT OF 21
? 4.68
NOT BAD--YOU ONLY MISSED BY 2.12597 %
SQUARE ROOT OF  21 IS 4.58258

WHAT IS THE SQUARE ROOT OF 52
? 7.57
NOT BAD--YOU ONLY MISSED BY 4.97701 %
SQUARE ROOT OF  52 IS 7.2111

WHAT IS THE SQUARE ROOT OF 27
? 5.15
NOT BAD--YOU ONLY MISSED BY .888204 %
SQUARE ROOT OF  27 IS 5.19615

THAT'S 3 IN A ROW!        *****
  PASTE STAR HERE--       *   *
                          *****
```

Notice that the user had to supply an answer within 5% three times in a row before getting the "gold star."

For a really fiendish game, make the 5% a variable that gets smaller each time. Start with V=.05, and then make V=.7*V each time around.



IBM stands behind *Creative Computing.* (Bob Taylor of Columbia University Teachers College, at a recent Paris meeting.)

REM

We have been explaining programs by drawing "balloons" on the side, which contain explanatory remarks. Remarks can also be placed within a program by use of the REM statement which looks like this:

10 REM ANYTHING YOU WANT TO SAY

Remark statements show up only when you list a program, *not* during a run. Here's an example of how one of our previous programs might look with REM statements. It also illustrates a feature in *some* BASICs that allow remarks after the ! or ' symbol. →

```
LIST

10 REM---PROGRAM FOR CHECKING INPUT----------------
15 PRINT "TYPE A POSITIVE INTEGER BETWEEN 50 AND 100"
20 INPUT A
25 REM---FIRST SEE IF MAYBE IT'S NOT AN INTEGER---
30 IF INT(A) <> A THEN 120
35 REM---NOW SEE IF IT'S OUTSIDE RANGE 50 TO 100---
40 IF ABS(75-A) > 25 THEN 140
50 PRINT "YOU HAVE OBEYED A COMPUTER."
60 PRINT "THERE IS NO HOPE."
70 STOP
115 REM---MESSAGE FOR LINE 30 BRANCH---          !
120 PRINT "THAT'S NOT AN INTEGER"                ! THIS IS THE
130 GOTO 10                                      ! SECTION OF
135 REM---MESSAGE FOR LINE 35 BRANCH---          ! THE PROGRAM
140 PRINT "OUT OF REQUESTED RANGE"               ! THAT PRINTS
145 PRINT "READ THE INSTRUCTIONS CAREFULLY"      ! MESSAGES
150 GOTO 10                                      !
160 END
```

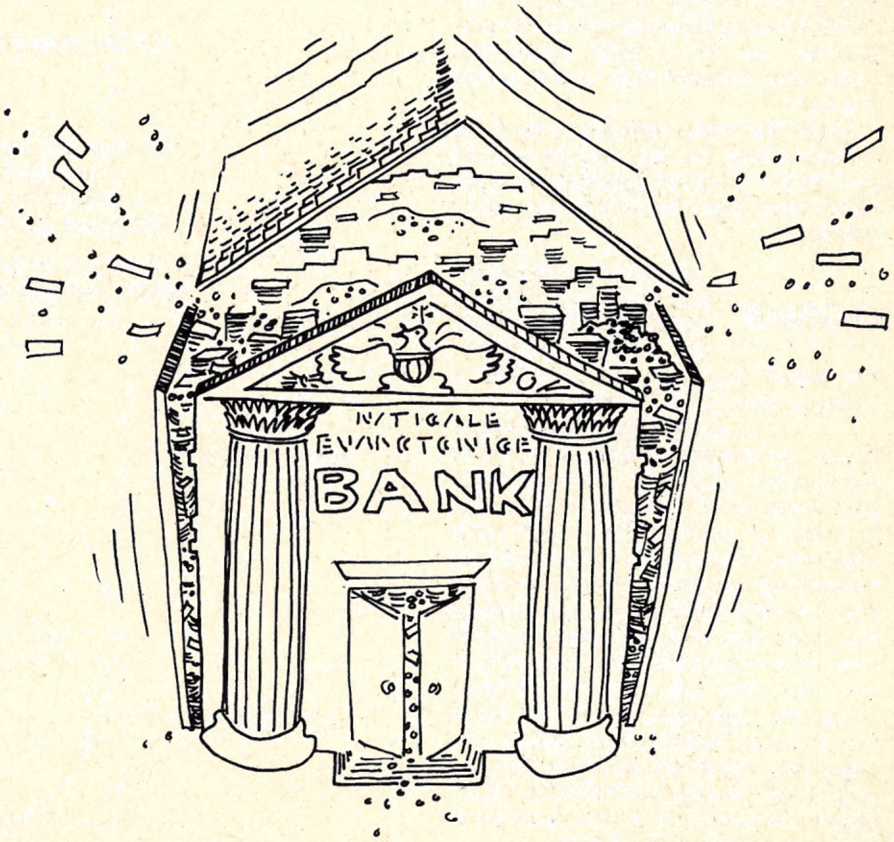
SELF-TEST

1. Simulate a RUN of this program:
```
10 FOR K = 1 TO 10
20 LET X = K * K
30 PRINT K, SQR(X)
40 NEXT K
50 END
```
2. Simulate a RUN of this "pattern" program:
```
10 FOR K = 10 TO -10 STEP -1
20 FOR J = 1 TO ABS(K)
30 PRINT "*";
40 NEXT J
50 PRINT
60 NEXT K
70 END
```
3. Write, debug, and run a program that asks for an estimate of the total cost of five items on a supermarket receipt. First have the computer print out the simulated receipt. Generate the dollar cost of each item with INT(900*RND(O) + 20)/100). Then ask for estimated total, and compare it with exact sum. Give different kinds of congratulatory (or other) messages that depend on the relative error in each answer.

2.8 HOUR 8: KEEPING CHECK ON A BANK BALANCE

In this section we'll explain two new features of BASIC (subroutines and user-defined functions) by showing how to apply them to the problem of calculating compound interest. Before discussing these features, let's first review what's involved in finding interest that's "compounded" at various intervals.

The idea of compounding shows up in several kinds of problems. For example, in calculating the population growth of some species (say, rabbits), you have to allow for the fact that if new rabbits come from the original population, then new, new rabbits come from both the new rabbits and the original population, while new, new, new rabbits come from the new,



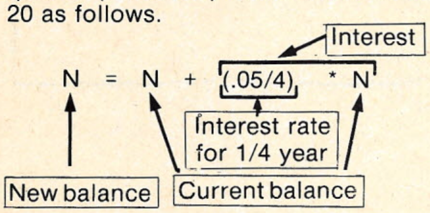
new rabbits, new rabbits, and original rabbits (assuming no deaths), etc., etc.

The same idea holds for compound interest: it's calculated on the original amount (called the principal), and on the interest on the principal, and on the interest on the interest on the principal, etc., etc. How often this re-calculation gets done is up to the bank. For example, they may do it four times a year (which is called quarterly compounding), or even 365 times a year (called daily compounding). There are two methods for calculating compound interest: (1) use a loop, and (2) use an exponential formula. Let's look at the loop method first.

Here's a loop for finding 5% interest compounded quarterly on a principal of $1000, with a total time in the bank of 1 year.

```
5 LET N = 1000
10 FOR K = 1 TO 4
20 LET N = N + (.05/4) * N
30 NEXT K
40 PRINT "INTEREST IS"; N - 1000
```

The new balance at the end of each quarter (3 months) is calculated in line 20 as follows.



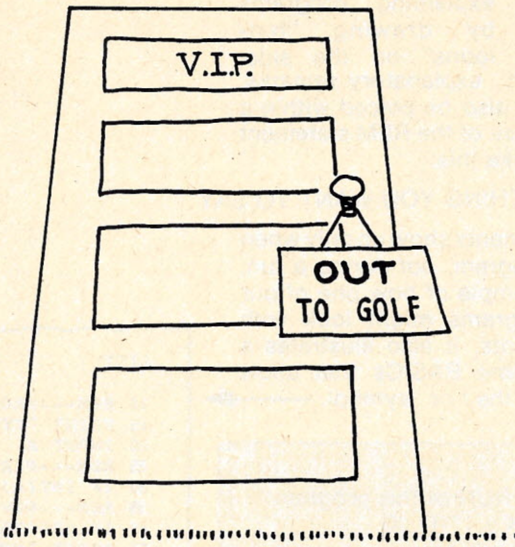
Each time around the FOR...NEXT loop is like another 3 months. At the end of 4 loops, N contains the year-end balance, so N - 1000 gives the compound interest that accumulated in a year.

To do this same calculation for daily compounding, the loop would have to go FOR K = 1 TO 365, while the interest added each day would be at the rate of (.05/365).

GOSUB

The small program we just explained can be used as part of a larger (or "main") program. The small program can be called a "subprogram", or a "subroutine."

The advantage to building a main program partially from subroutines is that it helps organize your thinking. The approach to take is to think of yourself as a VIP (very important programmer). You start by pretending that you don't have to worry about details because you can call on assistants for help. To make the idea even more dramatic, you can picture your executive office on the top floor, while the assistants work at lower levels called subroutines. When you need help from an assistant (say at level 1000) you shout "GOSUB 1000". When the assistant at this location is finished, he yells "RETURN".



This image isn't as silly as it may seem. To see why, let's first look at a "program" written by a VIP which only outlines the work to be done.

10 Get data on husband's bank account.

•
•
•

50 Get my assistant down on level 1000 to figure out and print husband's interest and balance.

•
•
•

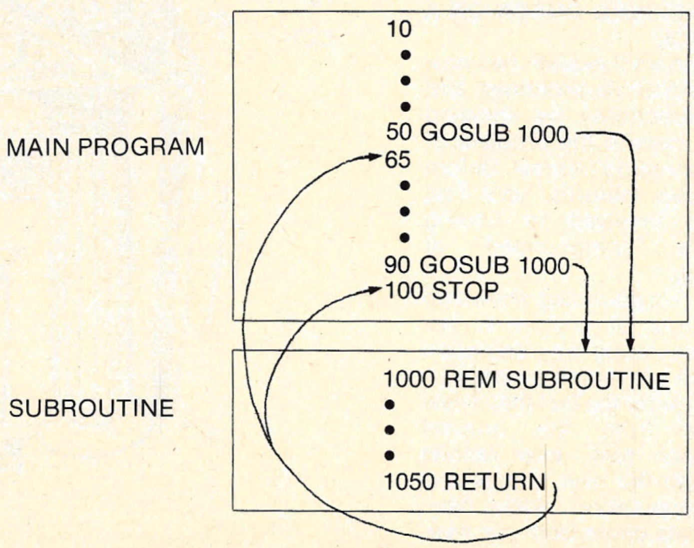
65 Get data on wife's bank account.

•
•
•

90 Ask the same assistant to figure out wife's interest and balance, and print it.

100 Lock up office and go play golf.

If written in BASIC, such a program would partially look like the following.



What the statement GOSUB 1000 really means is "go and do the subroutine that starts at line 1000 and then return to the line right after the GOSUB statement that was just executed." To see how this all goes together, look at the following complete program:

```
LIST

10 PRINT "ENTER INTEREST RATE";
20 INPUT R
30 PRINT "ENTER BALANCE FOR HUSBAND'S ACCOUNT";
40 INPUT B
50 GOSUB 1000
65 PRINT
70 PRINT "ENTER BALANCE FOR WIFE'S ACCOUNT";
80 INPUT B
90 GOSUB 1000
100 STOP
1000 REM SUBROUTINE FOR COMPOUNDING INTEREST DAILY OVER 365 DAYS
1005 LET N=B
1010 FOR D=1 TO 365
1020 LET N=N+(R/365)*N
1030 NEXT D
1040 LET I=N-B
1045 PRINT "YEAR'S INTEREST=";I;"NEW BALANCE=";N
1050 RETURN
1060 END


RUN

ENTER INTEREST RATE? .0525
ENTER BALANCE FOR HUSBAND'S ACCOUNT? 1000
YEAR'S INTEREST= 53.8986 NEW BALANCE= 1053.9

ENTER BALANCE FOR WIFE'S ACCOUNT? 1500
YEAR'S INTEREST= 80.8479 NEW BALANCE= 1580.85
STOP at line 100
```

All the hard work is done in the subroutine from lines 1000 to 1050. When the main program reaches line 50, it "goes to" line 1000, where it continues execution. In our example, it does line 1005, followed by 365 times around the FOR...NEXT loop in 1010 to 1030, followed by 1040, followed by 1045 and 1050. Line 1050 then says RETURN. (Subroutines must always end with a RETURN statement.) Return means go back to the line *right after* the GOSUB. In our example, that's line 65. So 65 is executed right after 1050. The second time the subroutine is called is at line 90. Again, all the hard work is done in the subroutine (at no extra cost in programming!), but this time the RETURN is to line 100.

Details, Details

Now that we see the big picture, we can concentrate on explaining how this particular subroutine works. What it does is to start the new balance out as N = B, calculate the interest for one day as (.0525/365)*N, and then get the revised new balance as N = N + (.0525/365)*N. This process is repeated 365 times in a loop. When the loop is finished, the interest earned for a year will be the final new balance minus the starting balance; that is, I = N - B. Now that we have N and I, we can return to line 65, where the program continues. When the program gets to line 90, this whole process is repeated, but this time B contains the wife's balance, so a completely different calculation is done. In other words, *subroutines in BASIC use the current value that variables have in the main program.*

Question: Could this program have been written as easily using GOTO instead of GOSUB? No, because there would be no way to return to different line numbers the way RETURN does.

Question: Can subroutines sometimes be inefficient? The answer is yes, but after everything is working, you can swap your VIP hat for your STP hat (super terrific programmer), and clean things up a bit. For example, the subroutine we have shown does the division (R/365) seven hundred and thirty times! This inefficiency can be removed by adding the statement

25 LET F = R/365

and using F instead of R/365 inside the subroutine.

DEF FNX

As just seen, subroutines are small programs, usually involving several lines. Sometimes a "subjob" can be handled by a single LET statement, and using a subroutine is hardly worth the effort. In this case, there's another feature called DEFining a function that can be used instead of GOSUB. We'll illustrate its use with the second method for calculating compound interest.

If you dig through some math books, you'll find the following formula for getting the new balance on an account with compound interest.

N = P * (1 + R/M) ↑ (M* T)

In this formula,

P is the starting principal in dollars,
R is the annual interest rate,
M is the number of times interest is compounded each year, and
T is the number of years left in the bank.

For example, for $3000 left for three years in a bank with 5% interest compounded monthly,

P = 3000
R = .05
M = 12
T = 3

So the new amount at the end of three years,

N = 3000 * (1 + .05/12) ↑ (12 *3). The "up arrow" means raise to that power (exponentiate), so this is a difficult calculation. It's time for a computer!

Our programming approach will be to place this formula in a special statement that allows the formula to be called upon as often as we wish. The way to "store" a formula like this in a BASIC program is to use the define function statement as follows:

10 DEF FNN (P, R, M, T) = P * (1 + R/M) ↑ (M * T)

Keyword part

A one-letter name you choose

List of "arguments" used in the function

An expression (or 'formula') that tells what to do with the "arguments"

The DEF FNN statement can be placed anywhere in a program, and FNN can be used anywhere that an expression can be used. Here's a program that uses our function twice, once in a LET statement, once in a PRINT statement. We called our function FNN. We could just as well have used names like FNA, FNB, FNC, ..., FNZ.

THIS IS THE WAY ↑ IS PRINTED ON MANY TERMINALS.

```
LIST

10 DEF FNN(P,R,M,T)=P*(1+R/M)^(M*T)
20 PRINT "YOU HAVE $1000 IN A BANK THAT GIVES"
25 PRINT "5% INTEREST, COMPOUNDED DAILY."
30 PRINT "HOW MANY YEARS DO YOU WISH TO LEAVE IT";
40 INPUT T
50 LET N=FNN(1000, .05, 365, T)
60 LET I=N-1000
70 PRINT "YOU'LL RECEIVE $";I;" INTEREST AND HAVE A BALANCE OF $";N
80 PRINT "FOR";2*T;"YEARS, BALANCE WOULD =";FNN(1000,.05,365,2*T)
100 END


RUN

YOU HAVE $1000 IN A BANK THAT GIVES
5% INTEREST, COMPOUNDED DAILY.
HOW MANY YEARS DO YOU WISH TO LEAVE IT? 3
YOU'LL RECEIVE $ 161.822  INTEREST AND HAVE A BALANCE OF $ 1161.82
FOR 6 YEARS, BALANCE WOULD = 1349.83
```

As shown in lines 50 and 80, when FNN is used (or "called"), it must be given arguments. Notice that these arguments can be replaced with constants, variables, or even expressions.

ON...GOSUB

This statement is similar to the ON...GOTO statement. It directs the program to go to different subroutines, depending on the value of the variable (or expression) right after the keyword ON

```
•
•
•
10 ON K GOSUB 1000, 1500, 2000
20 PRINT
•
•
•
```

means "if K + 1, go to subroutine 1000, return to line 20",
"if K + 2, go to subroutine 1500, return to line 20",
"if K = 3, go to subroutine 2000, return to line 20".

It's up to the programmer to make sure that K takes on only values that match the number of subroutines. If, for example, K became 4 in our example, standard BASIC would treat this as an error (some earlier versions treated this as a "default" and continued execution at the next line—line 20 in our example). For an example of ON...GOSUB, see SELF-TEST Question 3.

SELF-TEST

1. Simulate running this program:

```
10 PRINT "TYPE AN INTEGER FROM 1 TO 5"
20 INPUT I
30 IF INT(I)    I THEN 10
40 IF ABS(I-3)   2 THEN 10
50 PRINT "HOW DO   ";
55 LET N =  I
60 GOSUB 500
100 PRINT
110 PRINT "OR IS IT ALREADY POSSIBLE ";
120 LET N = 2 * I
130 GOSUB 500
140 STOP
500 FOR K = 1 TO N
510 PRINT "YOU KNOW ";
520 NEXT K
530 RETURN
540 END
```

2. Simulate running this program:

```
10 DEF FNA(R) = 3.1416 * R * R
20 FOR K = 1 TO 5
30 PRINT "FOR A RADIUS OF"; 10   K
40 PRINT "THE AREA OF A CIRCLE IS"; FNA(10   K)
50 NEXT K
60 END
```

3. Write and run this program

```
10 FOR K = 1 to 10
20 PRINT "MAY ";
30 LET X = INT(4 * RND(0) + 1)
40 ON X GOSUB 100, 200, 300, 400
50 PRINT "SING TO ";
60 LET Y = INT(4 * RND(0) + 1)
70 ON Y GOSUB 100, 200, 300, 400
75 PRINT
80 NEXT K
90 STOP
100 PRINT "AN IMPORTED SALAMI ";
110 RETURN
200 PRINT "YOUR FAITHFUL DOG ";
210 RETURN
300 PRINT "AN ENRAGED CAMEL ";
310 RETURN
400 PRINT "THE EASTER BUNNY "

410 RETURN
500 END
```

4. Write a program that compares the "loop" method with the "formula" method for getting compound interest. Show the results at the end of each year. A RUN should look like this:

```
PRINCIPAL? 1000
INTEREST RATE? .05
# OF TIMES COMPOUNDED PER YEAR? 365
# OF YEARS? 25
STARTING YEAR? 1976
THANK YOU
```

	LOOP METHOD		FORMULA METHOD	
YEAR	BALANCE	INTEREST	BALANCE	INTEREST
1976	1000	0	1000	0
1977	1051.27	51.2745	1051.27	51.2663
1978	1105.16	53.9032	1105.16	53.8945
1979	1161.82	56.6667	1161.82	56.6574
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
2001	3490.05	170.223	3489.94	170.191

Our program ignored leap years. If you're really ambitious, see if you can take leap years into account. The output of this program will vary slightly on different computers due to what are called "rounding" errors. The only way around this problem is to use a BASIC with double-precision arithmetic.

2.9 PROJECT IDEAS

1. Write a program that allows the user to enter the date of deposit, the amount deposited, the annual interest rate, the number of times compounded per year, and the date of withdrawal. The program should then print the new balance and the interest accumulated. A date like Nov. 18, 1976 can be entered as:

DATE DEP? 11, 18, 76

You can ignore leap years if you wish.

Another simplification is to treat all months as having 30 days, which means assuming 360 days for one year (some banking systems do this). Sub-project: How can a bank advertise that 5% interest compounded daily amounts to an annual interest rate of 5.47%?

2. Write an arithmetic practice program that uses four subroutines: one for addition problems, one for subtraction, one for multiplication, one for division. The RND function and ON...GOSUB should then be used to select the kind of problem (addition, subtraction, multiplication, or division) to be presented. Also try to use the method of SELF-TEST Question 3 to produce different kinds of messages for wrong answers, and other kinds of messages for correct answers. Here's what a RUN might look like:

```
ADDITION QUESTION: 5 + 6 = ? 11
RIGHT! YOUR REWARD WILL BE RICHES AND RIPE BANANAS.
SUBTRACTION QUESTION: 33 - 23= ? 16
WRONG—ANSWER IS 10
KEEP THIS UP AND YOU'LL FIND CHICKEN LIVERS IN YOUR SOCK
DIVISION QUESTION: ... etc. ...
```

"...We'd like to put two bucks on 'Happy Daddy' running in the seventh today..."

3. It's legal to have one subroutine call another subroutine in BASIC. The program below illustrates this feature. Study and run the program, and then write it *without* using GOSUB at all. Your program should produce the same output as shown in our example.

```
LIST

5 RANDOMIZE
10 PRINT "PLAYER #1 TYPE RANGE (0 TO 50)";
20 INPUT P
30 LET R = 50 * RND(0)
40 LET D1 = ABS(P - R)
50 GOSUB 1000
60 PRINT "PLAYER #2 TYPE RANGE (0 TO 50)";
70 INPUT P
80 LET R = 50 * RND(0)
90 LET D2 = ABS(P-R)
100 GOSUB 1000
110 IF D1 = D2 THEN 170
120 IF D1 < D2 THEN 150
130 PRINT "PLAYER #2 WINS"
140 GOTO 180
150 PRINT "PLAYER #1 WINS"
160 GOTO 180
170 PRINT " TIE SCORE"
180 GOTO 9999
1000 REM--------TARGET DISPLAY ROUTINE-----
1010 GOSUB 2000
1020 LET X = P
1025 PRINT "SHELL";
1030 GOSUB 3000
1040 GOSUB 2000
1050 LET X = R
1055 PRINT "U-BOAT";
1060 GOSUB 3000
1070 GOSUB 2000
1080 PRINT
1090 RETURN
2000 REM--------LINE ROUTINE-----
2010 FOR K=1 TO 60
2020 PRINT "-";
2030 NEXT K
2040 PRINT
2050 RETURN
3000 REM--------SHELL ROUTINE-----
3010 PRINT TAB(X+8);"<*>"
3020 RETURN
9999 END
```

4. Find a program written in an extended version of BASIC, and translate it into a version that runs on your system. The idea is to become familiar with the possibilities of extended BASIC so you can get a feel for those features you want to insist on when buying your next software package. It would also be a good idea to keep a notebook on the special features of your BASIC.

```
RUN

PLAYER #1 TYPE RANGE (0 TO 50)? 35
------------------------------------------------------------
SHELL                                        <*>
------------------------------------------------------------
U-BOAT                                            <*>
------------------------------------------------------------

PLAYER #2 TYPE RANGE (0 TO 50)? 10
------------------------------------------------------------
SHELL              <*>
------------------------------------------------------------
U-BOAT                                             <*>
------------------------------------------------------------

PLAYER #1 WINS
```

SOLUTION. We'll show a sample solution to this project as a guide to what's involved. Our solution will also help you to read programs written in BASIC-PLUS or EXTENDED BASIC. You'll see that most of the extensions can easily be translated into minimal standard BASIC, but at the cost of extra statements.

Our example will first show a Russian Roulette game program written in extended BASIC. Then we'll illustrate how each of the extended statements can be replaced by several simpler statements.

To illustrate how this extended BASIC program can be translated into minimal standard BASIC, we'll show the two versions side by side. As you'll note, most extended statements must be translated into several standard statements.

The numbers in circles on our diagram refer to the following six explanatory notes.

```
LIST

10 RANDOMIZE
20 PRINT "RUSSIAN ROULETTE": PRINT "---------------------"
30 PRINT "TYPE 1 TO SPIN CHAMBER, 0 TO QUIT"
35 N=0
40 INPUT "YOUR CHOICE IS";C
60 IF C=1 THEN PRINT "LOTSALUCK" ELSE PRINT "CHICKEN": GOTO 140
70 IF RND(0)>.85 THEN 100 ELSE N=N+1
80 IF N>=10 THEN 120 ELSE PRINT "--CLICK--"
90 PRINT: GOTO 40
100 PRINT "  BANG!!!  YOU'RE DEAD":  PRINT "SORRY ABOUT THAT"
110 PRINT:PRINT "NEXT VICTIM PLEASE":PRINT:GOTO 30
120 PRINT "YOU DID IT!!  10 MISSES! -- YOU WIN"
125 PRINT "YEA! ."; FOR K=1 TO 50
130 STOP
140 PRINT " GET SOMEONE ELSE WHO ISN'T SO SMART": PRINT:GOTO 30
150 END
```

Notes on the Translation from Extended BASIC to Minimal BASIC

1. Many extended BASIC's allow several statements on the same line

provided they are separated by colons. To translate, you merely write a separate line for each part. This is what we did with line 20. Other examples are shown in lines 90, 100, 110, 125, and 140.

```
LIST

10 RANDOMIZE
20 PRINT "RUSSIAN ROULETTE": PRINT "---------------------"
30 PRINT "TYPE 1 TO SPIN CHAMBER, 0 TO QUIT"
35 N=0
40 INPUT "YOUR CHOICE IS";C
60 IF C=1 THEN PRINT "LOTSALUCK" ELSE PRINT "CHICKEN": GOTO 140
70 IF RND(0)>.85 THEN 100 ELSE N=N+1
80 IF N>=10 THEN 120 ELSE PRINT "--CLICK--"
90 PRINT: GOTO 40
100 PRINT "  BANG!!!  YOU'RE DEAD":  PRINT "SORRY ABOUT THAT"
110 PRINT:PRINT "NEXT VICTIM PLEASE":PRINT:GOTO 30
120 PRINT "YOU DID IT!!  10 MISSES! -- YOU WIN"
125 FOR K=1 TO 10: PRINT "YEA! ";: NEXT K: PRINT
130 STOP
140 PRINT "GET SOMEONE ELSE WHO ISN'T SO SMART": PRINT: GOTO 30
150 END
```

```
LIST

10 RANDOMIZE
20 PRINT "RUSSIAN ROULETTE"
21 PRINT "---------------------"
30 PRINT "TYPE 1 TO SPIN CHAMBER, 0 TO QUIT"
35 LET N=0
40 PRINT "YOUR CHOICE IS";
41 INPUT C
60 IF C=1 THEN 63
61 PRINT "CHICKEN"
62 GOTO 140
63 PRINT "LOTSALUCK"
70 IF RND (0)>.85 THEN 100
71 LET N=N+1
80 IF N>=10 THEN 120
81 PRINT "--CLICK--"
90 PRINT
91 GOTO 40
100 PRINT "BANG!! YOU'RE DEAD"
101 PRINT "SORRY ABOUT THAT"
110 PRINT
111 PRINT "NEXT VICTIM PLEASE"
112 PRINT
113 GOTO 30
120 PRINT "YOU DID IT!! 10 MISSES! -- YOU WIN"
125 FOR K=1 TO 10
126 PRINT "YEA! ";
127 NEXT K
128 PRINT
130 STOP
140 PRINT"GET SOMEONE ELSE WHO ISN'T SO SMART"
142 PRINT
143 GOTO 30
150 END
```

2. Line 35 shows that many extended BASIC's allow you to omit the word LET.
3. Line 40 shows how a message can be placed within an extended INPUT statement. This translates into a PRINT followed by an INPUT, with the PRINT terminated by a semi-colon.
4. Line 60 shows how the THEN in an IF...THEN statement can be followed by another statement rather than a line number. The translation can be a bit tricky as shown, since ELSE is also used.
5. If...THEN...ELSE means if true, go to the statement after THEN, if false, go to the statement after ELSE.

Other features of extended BASIC will be introduced as they become useful. Techniques for translating them into minimal BASIC will also be shown.

WARNING: When you're finished with a translation make sure that all your "GOTO" and "IF...THEN" statements branch to the correct line numbers. You may have to make some changes.

Here's a sample RUN of the ROULETTE program to show how it should work if you've done the translation properly. Of course runs will differ with different RND generators (and RANDOMIZE routines).

```
LIST
RUSSIAN ROULETTE
---------------------
TYPE 1 TO SPIN CHAMBER, 0 TO QUIT
YOUR CHOICE IS? 1
LOTSALUCK
  BANG!!!  YOU'RE DEAD
SORRY ABOUT THAT

NEXT VICTIM PLEASE

TYPE 1 TO SPIN CHAMBER, 0 TO QUIT
YOUR CHOICE IS? 1
LOTSALUCK
--CLICK--

YOUR CHOICE IS? 1
LOTSALUCK
--CLICK--

YOUR CHOICE IS? 1
LOTSALUCK
--CLICK--

YOUR CHOICE IS? 1
LOTSALUCK
--CLICK--

YOUR CHOICE IS? 1
LOTSALUCK
--CLICK--

YOUR CHOICE IS? 1
LOTSALUCK
--CLICK--

YOUR CHOICE IS? 1
LOTSALUCK
--CLICK--

YOUR CHOICE IS? 1
LOTSALUCK
--CLICK--

YOUR CHOICE IS? 1
LOTSALUCK
--CLICK--

YOUR CHOICE IS? 1
LOTSALUCK
YOU DID IT!!  10 MISSES! -- YOU WIN
YEA! YEA! YEA! YEA! YEA! YEA! YEA! YEA! YEA! YEA!
STOP at line 130
```

EOF

This ends the four-part excerpt from Chapter 2 of a new book, "BASIC and the Personal Computer," by Thomas A. Dwyer, to be published early this year by Addison-Wesley, Co., Reading, Mass. 01867. ■

Illustration reprinted with permission from Hammermill Paper Co.

Human Memory Processes

Susan Hastings

(who also wrote the three subsequent articles)

In all human communication, information is transmitted from one person's memory to another. From the ancient days of Greece when philosophers drew on their own conscious experiences of mental activities in order to understand just how the mind operates, up until modern times when scientists use electronic tools to explore memory processes, man has wanted to know more about how messages are coded, preserved, and retrieved by the brain. And as more ways are developed for transmitting information, the study of human memory becomes increasingly important.

Experimental psychologists have investigated two different types of human memory, one like the temporary "scratch-pad" buffers of digital computers, and the other like their more permanent "bulk-storage" devices. Human *short-term memory* can hold between five and ten pieces of information. Unless a person repeats it to himself, the information is typically forgotten in less than a minute. This temporary storage system is used when we look up and dial a new telephone number or when we recall what someone said a moment ago.

In contrast, the capacity of *long-term memory* is tremendous. It normally contains records of enormous numbers of facts and experiences. Employing this essentially permanent storage system, a person can preserve information for many years without making any conscious effort to retain it. Long-term memory is used when we recall our own telephone number, recognize the meanings of words, or remember someone's face.

A technique pioneered at Bell Labs is providing a new way to study both long-term and short-term memory. Instead of employing traditional methods of measuring the number and pattern of errors people make under controlled conditions, the new technique measures how long it takes a person to remember selected kinds and amounts of information correctly. One interesting aspect of the findings of reaction-time experiments is that what actually happens in a person's mind when he remembers something may be very different from what he thinks is happening. Not only are people unaware of some of their mental activities, but they may actually be mistaken about others. This means that to determine the brain's capabilities and limitations, precise measurements must be taken.

For example, in one of the first experiments Bell scientists performed on short-term memory, it was found that mental recognition is a more exhaustive process than any of the people being tested believed. When electronic apparatus was used to measure how long it took people to decide whether a particular test digit was among those presented in a short memorized list of digits, most people were surprised to discover that they searched the entire list for the test digit rather than stopping as soon as they found it. When reaction times from the experiment were studied, it was discovered that each digit in the memorized list required about 40 milliseconds to recall, but people showed exactly the same reaction time whether the test digit was in the middle of the list, at the end, or not included in the list at all. Furthermore, the test concluded that an exhaustive search is actually more efficient than a deliberately self-terminating one; when people were presented with different experimental

tasks that made it important to stop searching as soon as the test digit was found, the search became much slower.

Reaction-time experiments were also used in the study of long-term memory processes. In this area, concentration centered on words rather than numbers: on how common words are organized in memory, and how information about words is retrieved during such activities as reading and listening. One set of findings indicates that words are arranged systematically rather than haphazardly in long-term memory. Evidence that the brain stores words in the same way that a thesaurus does is supported by the fact that a person can decide that a row of letters like DOCTOR is a word considerably faster when it follows a related word like NURSE than when it follows an unrelated word like BREAD. The increase in speed is greatest when the word follows a related word immediately, instead of after a delay; the reaction time is then shorter by as much as 50 to 100 milliseconds. It was also found that people can *pronounce* a printed word faster when it is related in meaning to an immediately preceding word than when it is unrelated.

While experiments have shown that short-term memory seems to work very much like a digital computer, the computer analogy does not fully carry over to long-term memory processes. For example, people were sometimes presented with a sequence of three words like NURSE-SMOKE-DOCTOR, in which two related words were separated by an intervening unrelated word. Despite the separation, people still recognized the third word more quickly than when the first word was unrelated to it, as in a sequence like BREAD-SMOKE-DOCTOR. This would not have happened with a magnetic-tape computer which proceeds its search from the point at which it has just stopped. Scientists believe that reaction timing shows that retrieving a word from long-term memory temporarily increases neural activity at the locations of their "nearby" related words in the memory structure. This "spreading activation" reduces the amount of additional activity needed to recognize a related word, and therefore speeds the reaction.

It is not too surprising that short-term and long-term memory processes operate in essentially different ways, because if a person had to go through a list of the thousands of words he is familiar with every time he spoke a sentence, it might take several minutes even if his memory worked at peak efficiency! But further experiments have shown that short-term and long-term memory search processes are stimulated in separate ways. While a short-term search may proceed only after the brain has mentally coded the stimulus into a highly refined, abstract representation (possibly the name of a test digit), long-term memory processes can proceed as soon as a person has a visual image of the stimulus.

Like the results of other basic research, findings about short-term and long-term memory can potentially be used in many ways. Results from memory research may suggest ways to improve communication by increasing a person's rate of proceesing information, or by reducing errors. Naturally the Bell researchers first related their findings to ways to make the telephone more efficient: they wish to assign frequently-called telephones numbers that can be remembered most easily, and make the job of a telephone operator minimally difficult. But recent discoveries about human memory have begun to influence research far removed from the topics originally investigated. Scientists are using reaction time experiments to study child development aging, the effects of acceleration in space and with hallucinogenic drugs, and the nature of aphasia, schizophrenia, and mental retardation. Reaction-time methods are being employed to attack a growing variety of other questions about mental processes, both fundamental and applied. During the past decade, the percentage of psychological studies involving reaction time measurements has nearly tripled.

For more information, see "Exploring the Speed of Mental Processes" by Max Mathews, David Meyer, and Saul Sternberg, *Bell Laboratories Record,* Vol. 53, No. 3, March 1975.

Mind-Reading Via Computer

In a recent French science fiction movie, *Fantastic Planet,* learning was accomplished by putting on aheadband that fed information directly into the brain. Today, a California neurophysiologist and psychologist is on his way toward making this imaginary device a reality.

For the past two years, Dr. Lawrence Pinneo of the Stanford Research Institute has been working with eight subjects to develop a means to record and store human thoughts in a computer. He calls this sytem, which would result in a closer coupling between man and machine, biocybernetic communication.

In Dr. Pinneo's experiments, electroencephalographic signals from a subject's brain are recorded when the subject speaks or thinks a specific word, and a computer stores the signals associated with that word. The next time the subject speaks or thinks the word, the computer sorts through its vocabulary, picks out the correct pattern, and responds by ordering a remote-control camera to perform the appropriate action.

Results from Pinneo's experiments have proven significantly successful. The fact that some words do have a higher percentage of correct classification than others has led him to believe that 100 percent effectiveness can be achieved as the process continues to be refined.

Doing that involves learning what is actually happening in the brain when someone speaks or thinks a word. Semantic components of words are being studied to see how much of a word's meaning is conveyed by its sound and how much is the result of some other unspecified process. As his studies progress, Pinneo has become convinced that there is a way to trap the essential semantic context of words and program them. If this can be done, the communication between man and machine could work in both directions. A computer which could store information in electrophysiological signals could also convey information via those signals to the human brain, and *Fantastic Planet's* teaching headband would be only one of the uses of biocybernetic communication.

In the future, thoughts transformed by computers into mechanical directions would make it possible for astronauts, aquanauts, or for that matter, even land-bound drivers of futuristic automobiles to maneuver their vehicles simply by thinking into a computer-linked engine. Artists could create by conjuring up images, in their minds, of the finished products. Even distant planets with hostile environments could be explored by using a two-way biocybernetic link, giving man the illusion of direct experience and the advantage of immediate, rather than pre-programmed thought.

The possibilities of biocybernetic communication are extraordinary and even frightening. Man's search for communion with others, even if in some cases the other may be a machine, is, as in all quests, both a hunt for something and an escape from something. Using Piinneo's mind-computer link, we may finally succeed in completely knowing each other and our environment, but we also may risk losing our private selves.

The Computer Meets the Doctor

Artificial Intelligence (AI) in medicine is perhaps one of the most disputed and exaggerated territories in computer research today. The computer as a consultant appears to be an almost certain fixture in the medical diagnosis of tomorrow, but neither experts nor laymen can seem to agree on just how far doctors can and should go in their reliance upon the judgement of a machine.

The term AI refers to computerized information processing tasks that have previously been assigned to the human brain, approaching the thin, hazy line between programming and learning, rote memory and intelligence. Characterized by a complex reasoning structure composed of informational and investigative models, high-level computer languages and computational methods which are primarily symbolic (non-numeric) in nature, the AI computer is capable of multiple modes of logic, data interpretation, therapy selection, prognostication, decision-making, and in some cases, self-improvement.

MYCIN, an AI program developed by Dr. Edward Shortliffe and others, is one of the electronic physicians that might soon find its way into many hospitals. The MYCIN system is designed to advise physicians about antibiotic therapy for patients with infectious diseases, especially before complete information about the organism which caused the disease becomes available. Although written in advanced computer dialect, MYCIN provides the doctor with an Explanation System that understands English questions, and can retrace its own reasoning pattern in order to make its human colleague aware of how the decision was made. New information can also be added to the MYCIN system through its Rule Acquisition System, so that a two-way communication between man and machine is always possible.

Numerous other AI projects designed for diagnosis and therapy are being tested in hospitals and clinics right now. Professor Casimir A. Kulikowski and Dr. Aran Safir are now experimenting with a project that would enable a computer to "examine" and treat a patient (a marvel forecast in Michael Crichton's *The Andromeda Strain*); they have already developed a computer model that analyzes and prescribes for patients with severe cases of glaucoma. Perhaps the most spectacular AI system now in use, however, is a computer alarm system which screens prescriptions at Latter Day Saints Hospital Pharmacy in Salt Lake City. The Drug Warning System evaluates a complete, automated file of a patient's history, lab data, previous drug therapy and disease state via a sophisticated AI logic system in order to assess the probability of circumstances under which a doctor's prescribed drug might become dangerous. The same AI system has also been used to take detailed medical histories and diagnose over 20,000 patients.

In the area of pure research, artificial intelligence is now being used to generate probable "candidate" molecular structures from mass spectrometry data, X-ray protein crystallography and organic synthesis. Computers that function like the brain are also being developed to help medical students learn to deal with psychological dysfunctions.

Perhaps, though, the biggest frontiers of AI are ethical in nature. Some AI proponents feel that if doctors look upon the computer as a tool, it will help eliminate many repetitive tasks and allow them to function more efficiently. Other experts doubt the validity of the term "artificial inteligence," and say that a way has not yet been found to make a computer capable of general problem solving and learning. Most experts do agree on one thing though. If the day ever comes when an artificial intelligence program has the capacity of acquiring new knowledge, the question of moral and legal responsibility for a computer's action will inevitably come to the fore.

Tidewater's Police Network

TENPIN (Tidewater Electronic Network of Police Information) is a telecommunications network designed to give police officers instant access to information from local and national computerized crime data files. What makes the TENPIN system unique is that it is one of the few regional groupings of independent municipalities (as opposed to a grouping under a country or metropolitan authority) that is improving law-enforcement effectiveness by pooling information about criminals in their respective jurisdictions through computer storage and retrieval.

There are 1,500 police officers in the greater Tidewater area of southern Virginia. The City of Norfolk's Data Processing Division developed TENPIN to help them protect the 1½ million people who live within its 1500 square miles.

TENPIN terminals are located at police headquarters in 10 cities and towns as well as at the U.S. Army's Fort Eustace and the FBI's Norfolk office. From anywhere in the TENPIN area a police officer can radio in a request for information to a terminal and receive a reply within a minute. TENPIN files are accessed through Norfolk's IBM 370/145, using 3270 and 2740 terminals for input and retrieval. These units in turn tie police into the National Criminal Investigation Commission (NCIC) computer and the FBI's file of wanted felons in Washington, D.C. In addition, the system links each TENPIN member to Virginia's Criminal Information Network (VCIN), a computer that connects them to the State Division of Motor Vehicles, the state police, other Virginia police departments, and the National Law Enforcement Teletype system (NLETS). Via the latter, each can send or receive a message to or from virtually any police department.

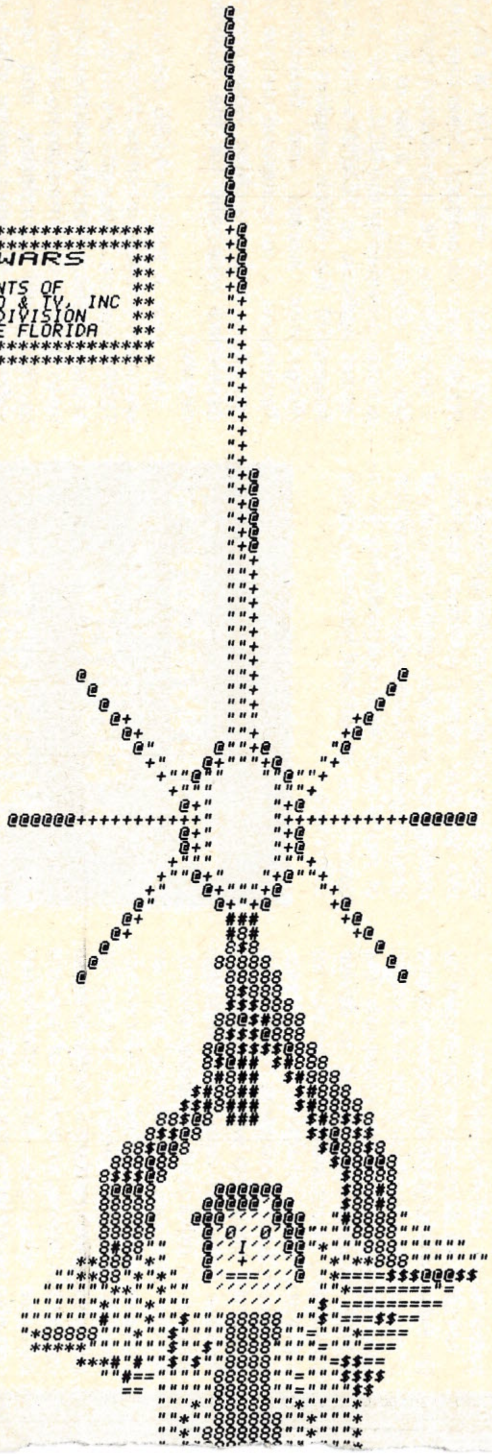
TENPIN includes six files, the main one being the warrant file of about 20,000 wanted persons. Other files are: a surveillance file, a probation/parole file, an alias fle, and a list of stolen aritcles provided by the NCIC. The most recent addition to TENPIN's files is a fingerprint file which employs FAST (Fingerprint Access and Searching Technique), a computerized method of recording and searching fingerprints developed especially for the TENPIN network.

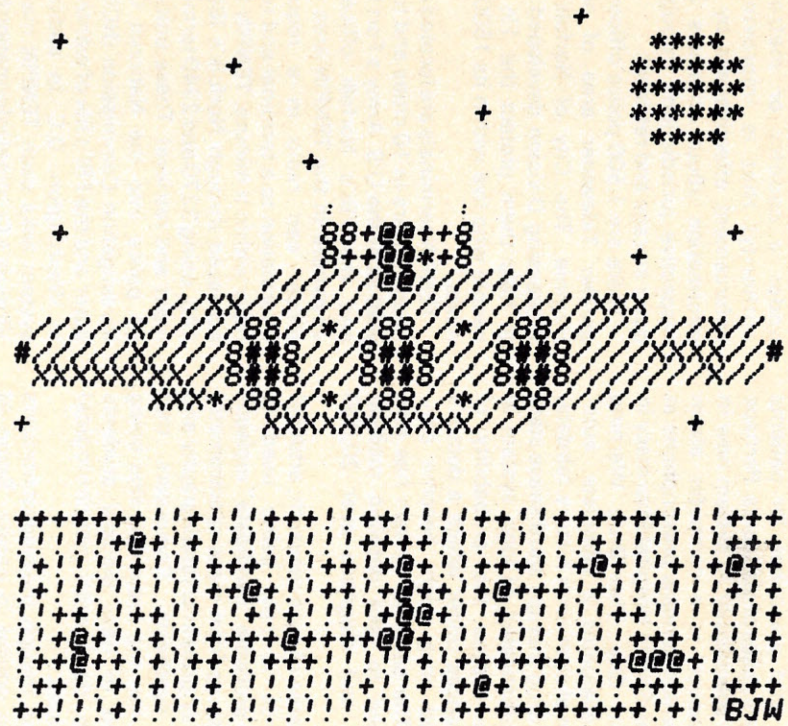
Last year a total of 4,096 warrants were served between cities through the TENPIN system. Police officers are proud of the statistic, and admit that they owe a lot to TENPIN. Criminals pay no attention to municipal boundaries, and law-enforcement agencies cannot affort to let those boundaries limit their effectiveness. As evidenced by TENPIN, a tool as efficient as the computer can help police to leap the boundaries far faster than the criminals can.

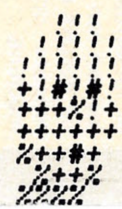
For more information, see "Tidewater's Police Network," *Data Processor,* March 1975. (IBM, DP Division, White Plains, NY 10604.) ■

STAR WARS

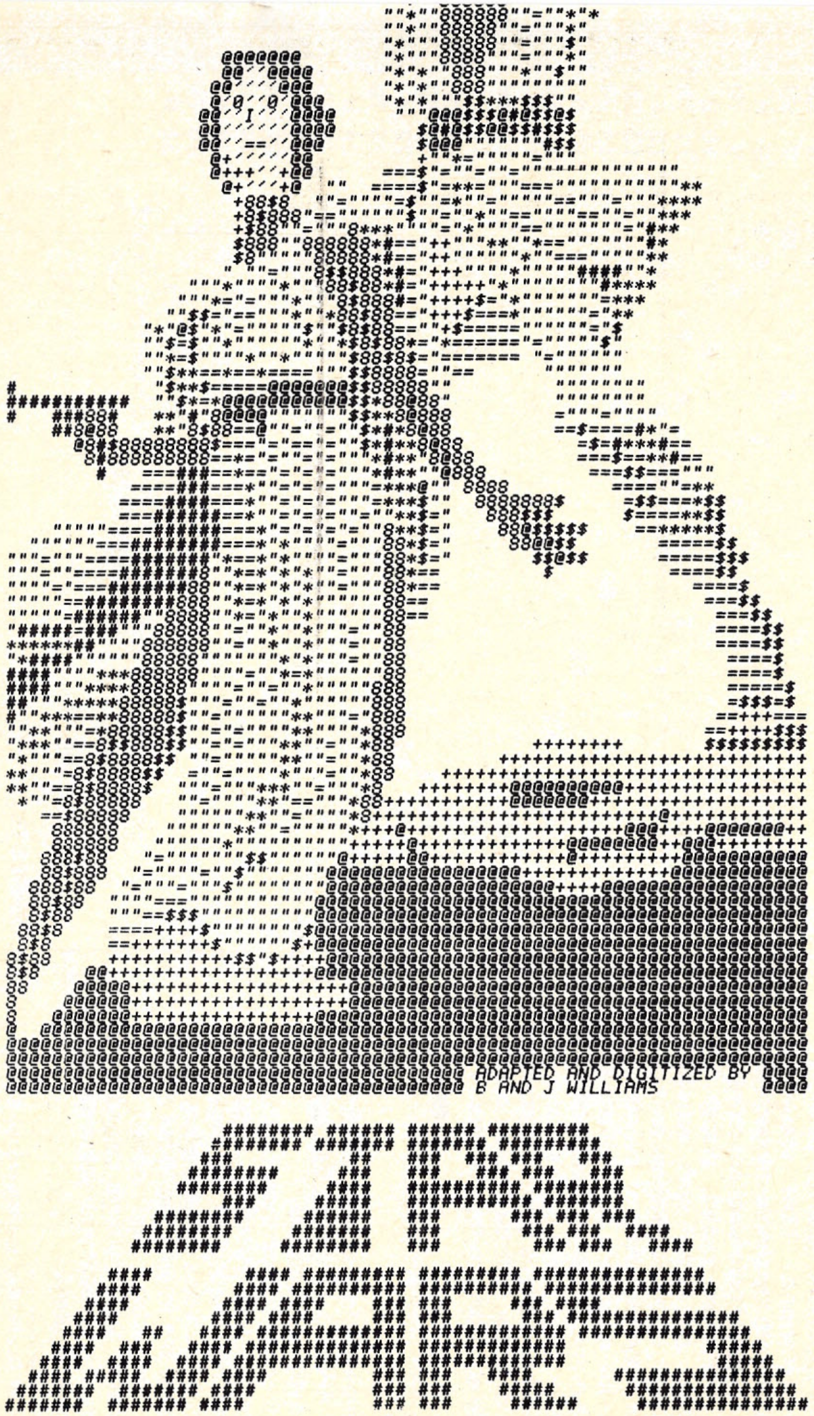
COMPLIMENTS OF
WILLIAMS RADIO & TV, INC
COMPUTER DIVISION
JACKSONVILLE FLORIDA

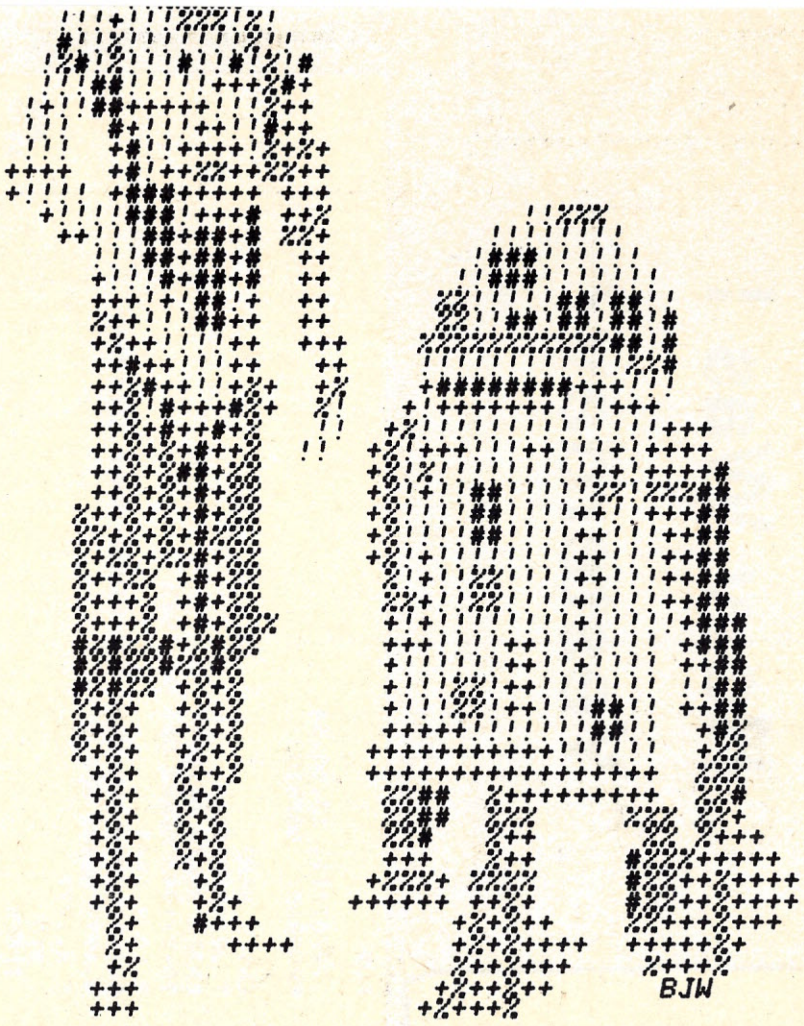






THE DROIDS


ADAPTED AND DIGITIZED BY
B AND J WILLIAMS
STAR WARS


BJW

How Was I Yesterday?

Yesterday was a disaster; an intellectual, physical, and emotional blight! I changed the password of my time-sharing account to a fruit that I cannot recall. I struck out (not once, but twice) in a slow-pitch softball game. I cried during a Daffy Duck cartoon on TV. How could this happen? Why me? The explanation is as close as my latest computer-generated biorhythmic calendar. My bio index was a paltry 927!

The theory of biorhythms, which has been around since the late 1800's, speculates that each person is guided by three cycles that begin at birth. The intellectual, physical, and emotional cycles are each sine curves with amplitude one and periods of 33, 23, and 28 days, respectively.

Biorhythms are open to a plethora of interpretations. That statement, coupled with the simple-minded structure of the periodic functions representing the three cycles, places biorhythms on a par with astrology. However, using biorhythms as a posteriori corroboration of events is a pleasant diversion. To facilitate this biorhythmic hindsight, a calendar with graphs of the three cycles and a numeric bio index is useful. Interpretation of a day with bio index i can be made according to the continuum:

$$1 \le i \le 99999$$

BLEAK — BRIGHT

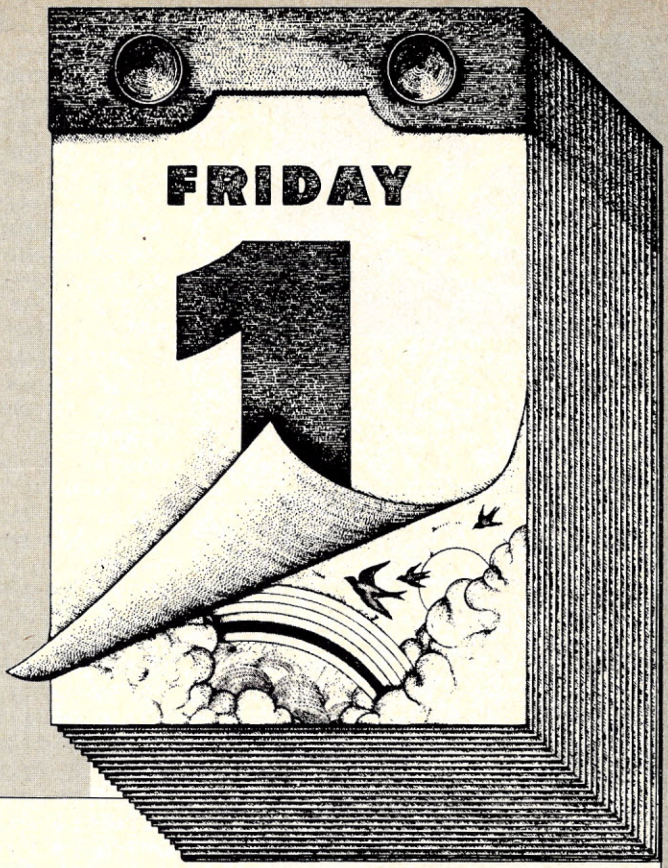
Below are interactions with BASBIO and APLBIO, programs that generate biorhythmic calendars.

It is apparent from these biorhythmic calendars that yesterday, August 19, found my intellectual, physical, and emotional cycles simultaneously low, yielding the inferior bio index of 927. My yesterday was indeed grim. Generate a biorhythmic calendar for yourself and see if your yesterday was gloomy or great.

*Dept. of Quantitative and Information Science, Western Illinois Univer., Macomb, IL 61455.

```
RUN

 77/07/11. 13.10.34.
PROGRAM   BASBIO

ENTER NAME
? JD ROBERTSON
ENTER BIRTH DATE
10,9,1943 MEANS OCT 9, 1943
? 10 9 1943
ENTER MONTH AND YEAR FOR BIORHYTHMIC CALENDAR
4,1977 MEANS APR 1977
? 8,1977
  BIO                                                   AUG
 INDEX                                                 1977
         -....................O....................+
 69380                   P    .        E          I    1  M
 76886                        P           E       I    2  TU
 83274                        .    P          E  I     3  W
 88107                        .         P      IE      4  TH
 91022                        .              I   E     5  F
 91764                        .           I     P E    6  SA
 90201                        .        I         EP    7  SU
 86337                        .     I           E P    8  M
 80316                        . I             E  P     9  TU
 72421                      I .           E   P        10 W
 63052                  I     .        E   P           11 TH
 52710               I        .   E   P                12 F
 41966            I           E  P                     13 SA
 31425         I          EP  .                        14 SU
 21694       I       EP       .                        15 M
 13340     I     PE           .                        16 TU
  6857    I   E               .                        17 W
  2632    IPE                 .                        18 TH
   927    PI                  .                        19 F
  1855    E I                 .                        20 SA
  5377     EP I               .                        21 SU
 11302      E  PI             .                        22 M
 19299        E    IP         .                        23 TU
 28917            E    I P    .                        24 W
 39613               E    I   P                        25 TH
 50786                    E   I    P                   26 F
 61812                        E   I     P              27 SA
 72081                        .   E  I       P         28 SU
 81037                        .        E I      P      29 M
 88208                        .           E I     P    30 TU
 93231                        .               I   P    31 W
         -....................O....................+
                       JD ROBERTSON

SRU      2.060 UNTS.

RUN COMPLETE.
```

or Biorhythmic Hindsight



J.D. Robertson*

```
LIST

 77/08/16. 15.34.31.
PROGRAM    BASBIO

00100 DIM P1(23),P2(23),E1(28),E2(28),I1(33),I2(33)
00110 DIM W$(7),L$(41),M$(12)
00120 DATA SU,M,TU,W,TH,F,SA
00130 DATA JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC
00140 FOR N=1 TO 23
00150 P2(N)=SIN(6.283185307*(N-1)/23)
00160 P1(N)=INT(21.5+20*P2(N))
00170 NEXT N
00180 FOR N=1 TO 28
00190 E2(N)=SIN(6.283185307*(N-1)/28)
00200 E1(N)=INT(21.5+20*E2(N))
00210 NEXT N
00220 FOR N=1 TO 33
00230 I2(N)=SIN(6.283185307*(N-1)/33)
00240 I1(N)=INT(21.5+20*I2(N))
00250 NEXT N
00260 FOR N=1 TO 7
00270 READ W$(N)
00280 NEXT N
00290 FOR N=1 TO 12
00300 READ M$(N)
00310 NEXT N
00320 FOR N=1 TO 41
00330 L$(N)=" "
00340 NEXT N
00350 PRINT "ENTER NAME"
00360 INPUT N$
00370 PRINT "ENTER BIRTH DATE"
00380 PRINT "10,9,1943 MEANS OCT 9, 1943"
00390 INPUT M,D,Y
00400 PRINT "ENTER MONTH AND YEAR FOR BIORHYTHMIC CALENDAR"
00410 PRINT "4,1977 MEANS APR 1977"
00420 INPUT M4,Y4
00430 GOSUB 01350
00440 M=M4
00450 D=1
00460 Y=Y4
00470 GOSUB 00980
```

```
00480 S1=J
00490 GOSUB 01350
00500 L1=31
00510 IF M4=12 THEN 00570
00520 GOSUB 01110
00530 S3=N3
00540 M=M4+1
00550 GOSUB 01110
00560 L1=N3-S3
00570 B=J-S1+1
00580 E=B+L1-1
00590 PRINT "  BIO";TAB(56);M$(M4)
00600 PRINT " INDEX";TAB(54);Y
00610 PRINT TAB(9);"-....................0";
00620 PRINT "....................+"
00630 V=0
00640 FOR I=B TO E
00650 V=V+1
00660 J3=I-1
00670 K1=INT(J3/23)
00680 K2=J3-(K1*23)+1
00690 K3=INT(J3/28)
00700 K4=J3-(K3*28)+1
00710 K5=INT(J3/33)
00720 K6=J3-(K5*33)+1
00730 P=P1(K2)
00740 Q=E1(K4)
00750 R=I1(K6)
00760 O=P2(K2)+E2(K4)+I2(K6)
00770 O=INT (16666*(O+3))+1
00780 L$(21)="."
00790 L$(P)="P"
00800 L$(Q)="E"
00810 L$(R)="I"
00820 PRINT O;TAB(10);
00830 FOR N=1 TO 41
00840 PRINT TAB(7);L$(N);
00850 NEXT N
00860 PRINT TAB(54);V;TAB(58);W$(N2)
00870 L$(P)=" "
00880 L$(Q)=" "
00890 L$(R)=" "
00900 N2=N2+1
00910 IF N2<8 THEN 00930
00920 N2=1
00930 NEXT I
00940 PRINT TAB(9);"-....................0";
00950 PRINT "....................+"
00960 PRINT TAB(23);N$
00970 STOP
00980 IF M<3 THEN 01020
00990 M1=M-2
01000 Y1=Y
01010 GO TO 01040
01020 M1=M+10
01030 Y1=Y-1
01040 C=INT(Y1/100)
01050 D1=Y1-(C*100)
01060 N4=INT((13*M1-1)/5)+D+D1+INT(D1/4)
01070 N=N4+INT(C/4)-2*C+77
01080 N1=INT(N/7)
01090 N2=N-(N1*7)+1
01100 RETURN
01110 Y2=INT(Y/4)
01120 Y3=Y-(Y2*4)
01130 IF Y3=0 THEN 01150
01140 GO TO 01250
01150 Y2=INT(Y/100)
01160 Y3=Y-(Y2*100)
01170 IF Y3=0 THEN 01190
01180 GO TO 01230
01190 Y2=INT(Y/400)
01200 Y3=Y-(Y2*400)
```

```
01210 IF Y3=0 THEN 01230
01220 GO TO 01250
01230 L1=1
01240 GO TO 01260
01250 L1=0
01260 N1=INT((3055*(M+2))/100)-91
01270 L=0
01280 IF M<3 THEN 01330
01290 IF L1=0 THEN 01320
01300 L=1
01310 GO TO 01330
01320 L=2
01330 N3=N1+D-L
01340 RETURN
01350 IF M<3 THEN 01390
01360 M1=M-3
01370 Y1=Y
01380 GO TO 01410
01390 M1=M+9
01400 Y1=Y-1
01410 C=INT(Y1/100)
01420 D1=Y1-(C*100)
01430 N=INT((146097*C)/4)+D+INT((1461*D1)/4)
01440 J=N+1721119+INT((153*M1+2)/5)
01450 RETURN
01460 END
READY.
```

```
      APLBIO
ENTER NAME
JD ROBERTSON
ENTER BIRTH DATE
10 9 1943 MEANS OCT 9, 1943
⎕:
      10 9 1943
ENTER MONTH AND YEAR FOR BIORHYTHMIC CALENDAR
4 1977 MEANS APR 1977
⎕:
      8 1977
  BIO                                                      AUG
  INDEX                                                    1977
         -......................0......................+
  69380                    P    .         E           I   01 M
  76886                         P            E        I   02 TU
  83274                         .     P           E  I    03 W
  88107                         .          P       IE     04 TH
  91022                         .               I    E    05 F
  91764                         .            I      P E   06 SA
  90201                         .         I         EP    07 SU
  86337                         .      I            E P   08 M
  80316                         . I               E  P    09 TU
  72421                       I .            E    P       10 W
  63052                   I     .         E   P           11 TH
  52710                I        .    E   P                12 F
  41966            I            E  P                      13 SA
  31425         I           EP  .                         14 SU
  21694       I        EP       .                         15 M
  13340     I     PE            .                         16 TU
  06857    I   E                .                         17 W
  02632    IPE                  .                         18 TH
  00927    PI                   .                         19 F
  01855    E I                  .                         20 SA
  05377     EP I                .                         21 SU
  11302      E  PI              .                         22 M
  19299        E    IP          .                         23 TU
  28917            E     I P    .                         24 W
  39613                E    I   P                         25 TH
  50786                     E   I     P                   26 F
  61812                         E    I     P              27 SA
  72081                         .    E  I       P         28 SU
  81037                         .         E I       P     29 M
  88208                         .            E I      P   30 TU
  93231                         .                 I   P   31 W
         -......................0......................+
                           JD ROBERTSON
```

```
      ∇APLBIO[⎕]∇
      ∇APLBIO;M;N;L;V;K;W;P;E;I;O;S;LINE
[1]    'ENTER NAME'
[2]    W←⍞
[3]    'ENTER BIRTH DATE'
[4]    '10 9 1943 MEANS OCT 9, 1943'
[5]    M←⎕
[6]    'ENTER MONTH AND YEAR FOR BIORHYTHMIC CALENDAR'
[7]    '4 1977 MEANS APR 1977'
[8]    N←⎕
[9]    L←31
[10]   →SKIP×⍳N[1]=12
[11]   L←(IDAY(N[1]+1),1,N[2])-IDAY N[1],1,N[2]
[12]  SKIP:S←(IJLN N[1],1,N[2])-IJLN M
[13]   V←¯1+S+⍳L
[14]   P←1○((○2)×23|V)÷23
[15]   E←1○((○2)×28|V)÷28
[16]   I←1○((○2)×33|V)÷33
[17]   '    BIO';49⍴' ';(12 3
⍴'JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC')[N[1];]
[18]   '  INDEX',(47⍴' '),'0123456789'[1+10 10 10 10⊤N[2]]
[19]   '          -',(20⍴'.'),'0',(20⍴'.'),'+'
[20]   K←1
[21]  OUTPUT:LINE←41⍴' '
[22]   O←⌈16666×3+P[K]+E[K]+I[K]
[23]   LINE[21]←'.'
[24]   LINE[21+⌊0.5+20×P[K]]←'P'
[25]   LINE[21+⌊0.5+20×E[K]]←'E'
[26]   LINE[21+⌊0.5+20×I[K]]←'I'
[27]   '  ',('0123456789'[1+(5⍴10)⊤O]),'   ',LINE,'   ',
('0123456789'[1+10 10⊤K]),' ',,(7 2⍴'SUM TUW THF SA')[1+IZLR N[1
],K,N[2];]
[28]   →OUTPUT×⍳L≥K←K+1
[29]   '          -',(20⍴'.'),'0',(20⍴'.'),'+'
[30]   ((31-⌈0.5×⍴W)⍴' '),W
      ∇
```

```
      ∇IZLR[⎕]∇
      ∇R←IZLR N;M;C;D;Y
[1]    M←((N[1]<3),(N[1]≥3))/(N[1]+10),N[1]-2
[2]    Y←N[3]-N[1]<3
[3]    C←⌊Y÷100
[4]    D←100|Y
[5]    R←⌊(¯1+13×M)÷5
[6]    R←R+N[2]+D+⌊(D÷4)+⌊(C÷4)-2×C
[7]    R←7|R
      ∇
      ∇IJLN[⎕]∇
      ∇R←IJLN N;M;C;D;Y
[1]    M←((N[1]<3),(N[1]≥3))/(N[1]+9),N[1]-3
[2]    Y←N[3]-N[1]<3
[3]    C←⌊Y÷100
[4]    D←100|Y
[5]    R←1721119+N[2]+(⌊(146097×C)÷4)+(⌊(1461×D)÷4)+⌊(2+153×M)÷5
      ∇
      ∇IDAY[⎕]∇
      ∇R←IDAY N;M;L
[1]    M←0 31 59 90 120 151 181 212 243 273 304 334
[2]    L←ILEAP N[3]
[3]    R←M[N[1]]+N[2]+L×N[1]>2
      ∇
      ∇ILEAP[⎕]∇
      ∇R←ILEAP X
[1]    R←1=2|+/0=4 100 400|X
      ∇
```

Grammar as a Programming Language

Neil Rowe*

1. Introduction

A number of interesting projects involving such things as computer-generated sentences, poems, and stories have been presented recently in this magazine.[1] To perhaps clarify the similarities underlying these projects, I would like to propose a framework useful for such projects and others like them. This framework is the notion of a generative grammar. While grammars are usually associated with linguistics, their usefulness goes far beyond just "language" to many different domains. Their application is general enough to make grammars a sort of programming language in their own right.

A simple grammar-running control structure is presented, uncomplicated and very suitable for student tinkering. So not only can students write grammars, but they can modify and improve the grammar interpreter itself, learning something about how a simple kind of computer parser works.

2. A one-command computer language

Consider the control structure as a one-command computer language. (See Appendix for details of an implementation in the language Logo[2,3].) The R ("rule" or "replace") command can be expressed as a function (or procedure) with two list arguments:

R [NAME] [JOE]

The command works always on a single list (string) of words. It replaces in this list particular words by other particular words. That above command, for instance, says to look for all occurrences of the word NAME in the list and change them to JOE. We can also replace one word by several:

R [NAME] [THE PRESIDENT OF THE UNITED STATES]

Oftentimes you're going to want to make choices when replacing stuff. That is, you won't always want to replace NAME with the same name JOE. We might want for variety to replace it occasionally with TOM, DICK, or HARRY. We can write:

R [NAME] [(JOE TOM DICK HARRY)]

The parentheses mean for the computer to choose only one of the things inside them.

We can combine the brackets and parentheses in commands:

R [NAME] [(JOE TOM DICK HARRY) C. (JONES DOE DOAKES)]

which replaces NAME by JOE C. JONES, or HARRY C. DOE, etc. Or we can put brackets within parentheses:

R [NAME] [(JOE [TERRIBLE TOM] [DICK THE INSURANCE SALESMAN] HARRY) C. (JONES DOE DOAKES)]

Remember, brackets mean use everything inside them; parentheses mean choose one and only one of the things inside them.

*Artificial Intelligence Laboratory, Massachusetts Institute of Technology

3. Writing sentences

R commands by themselves aren't too interesting. You have to apply several in sequence -- a "grammar". Here's a way to create simple English sentences:

```
TO SIMPSENTENCE
10 R [SIMPSENTENCE] [NOUNPHRASE VERBPHRASE]
20 R [VERBPHRASE] [VERB NOUNPHRASE]
30 R [NOUNPHRASE] [([DETERMINER ADJECTIVE NOUN] NAME)]
100 R [VERB] [(LIKES HATES BOTHERS BEFRIENDS)]
110 R [DETERMINER] [(A THE SOME)]
120 R [ADJECTIVE] [(BIG TINY CHEERFUL SAD HAPPY GREEN PERPLEXED)]
130 R [NOUN] [(BOY GIRL COMPUTER ROBOT MARTIAN)]
140 R [NAME] [(TOM DICK HARRY SALLY SUE SANDRA)]
END
```

which can generate sentences like the following:

```
THE CHEERFUL ROBOT BOTHERS SOME SAD GIRL
TOM LIKES THE GREEN COMPUTER
A HAPPY MARTIAN BEFRIENDS SALLY
```

The nice thing is that you can add new features quite easily to this sentence generator. For instance, you can get sentences like

```
SOME BIG BOY IS SAD
A CHEERFUL ROBOT IS PERPLEXED
SANDRA IS CHEERFUL
```

by just changing line 20 to:

```
20 R [VERBPHRASE] [([VERB NOUNPHRASE] [IS ADJECTIVE])]
```

And you can include adverbs in your sentences. Change line 20 to:

```
20 R [VERBPHRASE] [ADVERB ([VERB NOUNPHRASE] [IS ADJECTIVE])]
```

and add a line 90:

```
90 R [ADVERB] [(OFTEN SURPRISINGLY PERHAPS DUTIFULLY)]
```

Examples:

```
DICK OFTEN IS HAPPY
A GREEN GIRL DUTIFULLY BEFRIENDS THE BIG ROBOT
```

4. Cyclic rule application

Suppose we want to allow an indefinite number of adjectives in front of the noun, like

THE BIG HAPPY GREEN ROBOT

which we might expect to do by

```
30 R [NOUNPHRASE] [([DETERMINER ADJSTRING NOUN] NAME)]
35 R [ADJSTRING] [ADJECTIVE (ADJSTRING [])]
```

(The bracket pair [] means the "empty list" or the "list of no elements". If it is chosen instead of ADJSTRING, ADJSTRING will be replaced by only ADJECTIVE.) The intent of line 35 is to just keep piling up adjectives in front of the noun until the random choice manages to choose the second choice.

But line 35 won't work because the both arguments contain the word ADJSTRING. Thus when you are done with line 35, there may still be ADJSTRING's in the string, and that's not what we want. What we really want then is to apply rule 35 again. One way to do this is just to apply the rules repeatedly, until on any particular time thru the rules you haven't been able to make any substitutions. Then you're done. (See the Appendix.)

Making the necessary changes to the control structure, we can now get:

```
THE GREEN PERPLEXED ROBOT SURPRISINGLY LIKES SOME TINY SAD BOY
SUE PERHAPS HATES A BIG GREEN HAPPY MARTIAN
```

Now suppose we want to have compound sentences, sentences composed of two subsentences

joined by a word like "and". Change line 10 to read

```
10 R [SIMPSENTENCE] [NOUNPHRASE VERBPHRASE ([] [(AND BUT SINCE THOUGH)
SIMPSENTENCE])]
```

giving:

THE TINY HAPPY ROBOT OFTEN LIKES SANDRA AND TOM SURPRISINGLY IS SAD

and so on. There are many possibilities for further development.

So here's our new improved sentence generator:

```
TO SIMPSENTENCE
10 R [SIMPSENTENCE] [NOUNPHRASE VERBPHRASE ([] [(AND BUT SINCE THOUGH)
SIMPSENTENCE])]
20 R [VERBPHRASE] [ADVERB ([VERB NOUNPHRASE] [IS ADJECTIVE])]
30 R [NOUNPHRASE] [([DETERMINER ADJLIST NOUN] NAME)]
35 R [ADJLIST] [ADJECTIVE ([] ADJLIST)]
40 R [VERB] [(LIKES HATES BOTHERS BEFRIENDS)]
50 R [DETERMINER] [(THE A SOME)]
60 R [ADJECTIVE] [(BIG TINY HAPPY SAD GREEN PERPLEXED)]
70 R [NOUN] [(BOY GIRL COMPUTER ROBOT MARTIAN)]
80 R [NAME] [(TOM DICK HARRY SALLY SUE SANDRA)]
90 R [ADVERB] [(OFTEN SURPRISINGLY PERHAPS DUTIFULLY)]
END
```

5. Writing poetry and the R.ONCE feature

Consider the problem of writing poems in which the last syllables of the line must rhyme. We could try

```
TO LIMERICK
10 R [LIMERICK] [A A B B A]
20 R [A] [DOWN UP DOWN UP DOWN RHYME1]
30 R [B] [DOWN UP DOWN RHYME2]
40 R [RHYME1] [(DATE FATE WAIT SATE SMELL BELL HELL WELL BROKE COKE JOKE YOKE)]
50 R [RHYME2] [(FEAT BEAT SEAT HEAT WAY SAY BAY HAY MOOD FOOD STEWED RUDE)]
60 R [DOWN] [(UH AH ER IR AN UN IS US AW E)]
70 R [UP] [(MEAN PROTE VAST SPRILL TRAMMED SLOOSED POUNT GRASP DRUNK)]
END
```

but this runs into a problem: each time the interpreter substitutes for RHYME1 or RHYME2, it will choose a new word. That is, we have no way to force a rhyme.

It seems what we need is a "new R command", call it R.ONCE, that works just like the old, except it only chooses once per cycle. (See the Appendix.) Using it we can rewrite our limerick-writing program this way:

```
TO LIMERICK
10 R [LIMERICK] [A A B B A]
20 R [A] [DOWN UP DOWN UP DOWN RHYME1]
30 R [B] [DOWN UP DOWN RHYME2]
40 R.ONCE [RHYME1] [(ATE ELL OKE)]
50 R.ONCE [RHYME2] [(EAT AY OOD)]
100 R [ATE] [(DATE FATE WAIT SATE)]
110 R [ELL] [(SMELL BELL HELL WELL)]
120 R [OKE] [(BROKE COKE JOKE YOKE)]
130 R [EAT] [(FEAT BEAT SEAT HEAT)]
140 R [AY] [(WAY SAY BAY HAY)]
150 R [OOD] [(MOOD FOOD STEWED RUDE)]
200 R [DOWN] [(UH AH ER IR AN UN IS US AW E)]
210 R [UP] [(MEAN PROTE VAST PRILL TRAMMED SLOOSED POONT GRASP DRUNK)]
END
```

A sample limerick:

AH GRASP UN POONT E DATE
AN SLOOSED IR POONT UH SATE
ER VAST AN FOOD
UN PRILL IS STEWED
AW PROTE IR TRAMMED US FATE

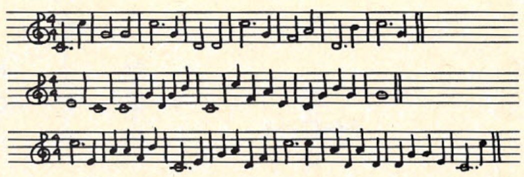
6. Writing music

Here's a program that writes melodies according to a few simple harmonic schemes. It first

chooses a harmony for each measure, then constructs measures to fit that harmony. To make its melody a little more unified, it uses R.ONCE to make sure that measures with the same harmony have the same rhythm. The final string consists of pitches (C, D, E, F, G, A, B, or CC) alternating with durations (Q (quarter), H (half), DH (dotted half), or W (whole) notes).

```
        TO MELODY
        10 R [MELODY] [CCHORD (CCHORD GCHORD FCHORD) CCHORD ([CCHORD GCHORD DCHORD]
[GCHORD DCHORD GCHORD] [GCHORD CCHORD FCHORD]) GCHORD CCHORD]
        20 R.ONCE [CCHORD] [([CNOTE W] [CNOTE DH CNOTE Q] [CNOTE H CNOTE Q CNOTE Q])]
        30 R.ONCE [GCHORD] [([GNOTE H GNOTE H] [GNOTE Q ORDNOTE Q GNOTE Q ORDNOTE Q])]
        40 R.ONCE [FCHORD] [([FNOTE H FNOTE H] [FNOTE Q ORDNOTE Q FNOTE Q ORDNOTE Q])]
        50 R.ONCE [DCHORD] [([DNOTE H DNOTE H] [DNOTE Q ORDNOTE Q DNOTE Q ORDNOTE Q])]
        100 R [CNOTE] [(C E G CC)]
        110 R [GNOTE] [(D G B)]
        120 R [FNOTE] [(C F A CC)]
        130 R [DNOTE] [(D F A)]
        140 R [ORDNOTE] [(D E F G A B)]
        END
```

Sample melodies are given in Fig. 1.



7. Drawing a robot face

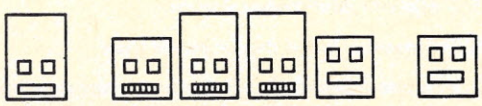
We can apply the idea of a grammar to drawing designs too. Consider the problem of drawing with a "turtle" that can either move forward a certain amount (abbreviated FD) or turn right a specified number of degrees (abbreviated RT). When moving forward, he can either leave a line behind him (PENDOWN mode) or not (PENUP mode). So to draw a rectangle:

```
FD 10 RT 90 FD 20 RT 90 FD 10 RT 90 FD 20 RT 90
```

Let's write a grammar to draw "robot" faces. We'll use a large square for the head, small squares for the eyes, and small rectangles in a row for the teeth. We make choices as to whether to have a tall head or a square head, have the eyes and mouth high or low in the head, and whether to show the teeth in the mouth.

```
TO FACE
10 R [FACE] [HEAD (SETHIGH SETLOW) EYES MOUTH]
20 R [HEAD] [(TALLHEAD SQUAREHEAD)]
100 R [EYES] [EYE SETUPEYE2 EYE]
200 R [MOUTH] [SETUPMOUTH (OPENMOUTH SIXTEETH)]
210 R [SIXTEETH] [TOOTHPLUS TOOTHPLUS TOOTHPLUS TOOTHPLUS TOOTHPLUS TOOTHPLUS]
220 R [TOOTHPLUS] [TOOTH PENUP FD 10 PENDOWN]
300 R [SETHIGH] [PENUP FD 80 RT 90 FD 20 PENDOWN]
310 R [SETLOW] [PENUP FD 50 RT 90 FD 20 PENDOWN]
320 R [SETUPEYE2] [PENUP FD 40 PENDOWN]
330 R [SETUPMOUTH] [PENUP FD 19 RT 90 FD 55 RT 90 PENDOWN]
END
```

where the remaining undefined words are just rectangles and squares of the appropriate size. Sample faces are given in Fig. 2.



8. Drawing hydrocarbons

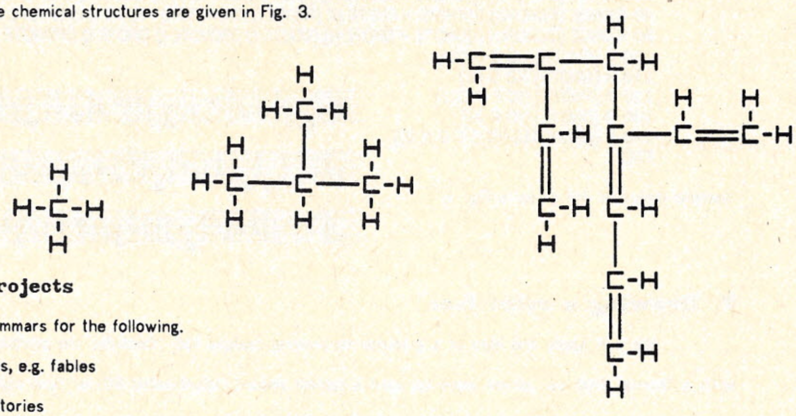
We can use grammars to explore some aspects of chemistry. Consider the following grammar for drawing some hydrocarbons:

```
        TO HYDROCARBON
        10 R [HYDROCARBON] [MARK "C" CHAIN LT 90 CHAIN LT 90 CHAIN LT 90 CHAIN LT 90]
        20 R [CHAIN] [(HATOM HATOM HATOM HATOM HATOM CATOM C2ATOM)]
        30 R [HATOM] [SHORTDASH MARK "H" RT 180 SHORTDASH]
        40 R [CATOM] [DASH MARK "C" LT 90 CHAIN LT 90 CHAIN LT 90 CHAIN LT 90 DASH]
        50 R [C2ATOM] [DASH MARK "C" LT 90 CHAIN LT 90 DOUBLEDASH MARK "C" LT 90 CHAIN LT 90
CHAIN DOUBLEDASH DASH]
        END
```

where MARK is a command that draws a letter, and where DASH, SHORTDASH, and DOUBLEDASH are defined in the obvious way.

The grammar builds up a molecule by starting with a carbon atom and attaching to each of its four sides either a hydrogen atom, a single-bonded carbon atom, or a double-bonded carbon atom pair. In the case of the last two, the process is repeated for bonds of those carbon atoms. Note that since the entire molecule must be drawn by a step-by-step process, "backing up" must be done at times.

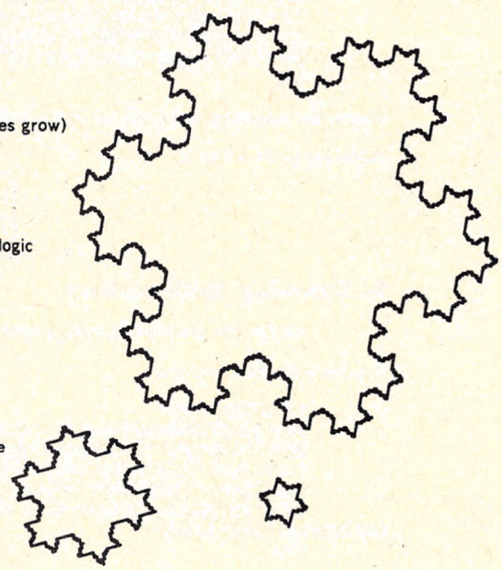
Some sample chemical structures are given in Fig. 3.



9. Additional projects

Try writing grammars for the following.

1. Simple stories, e.g. fables
2. Newspaper stories
3. Advertisements
4. New words (that sound like English, e.g., not like "ggnxyqprt")
5. Something like SIMPSENTENCE but with subject-verb agreement in number (singular vs. plural)
6. Or ability to use "an" and "a" properly
7. Or ability to substitute a pronoun (the correct one) occasionally
8. Sentences in some foreign language
9. Musical melodies based on melodic (rather than harmonic) considerations
10. Passacaglias on a given ground (see a music theory book)
11. Rondos where the sections are all in ternary form (ditto)
12. Apartment houses of random size and shape, with shades drawn for random windows, plants in the window for random windows, etc.
13. "Snowflake" curves (see Fig. 4)
14. Trees and bushes (find out something about how real ones grow)
15. A different kind of chemical structure
16. The physics of bubble chamber particle tracks
17. Some kind of electronic circuit diagrams, perhaps digital logic
18. Numbers which are powers of 2
19. Composite (not prime) numbers
20. The English names of numbers up to 100 billion
21. "Agendas" for your daily activities
22. Computer programs in some particular computer language



10. Further control structure modifications

As you may observe, one of the nicest things about this system is the relative ease of making changes to the control structure (or interpreter), thanks to its relative simplicity. Feel free to experiment. Do some reading about linguistics.[4,5,6]

For one, it might be nice to have context-sensitivity of a sort: have rules that replace a string of words by a string of words. For instance, we could generate Fibonacci numbers by the following:

```
TO FIBO
10 R [FIBO] [A + B]
20 R.ONCE [A] [(1 NEWB)]
30 R [1 + B] [1 + 1]
40 R [B + 1] [1 + 1]
50 R [B] [FIBO]
60 R [NEWB] [B]
END
```

The resulting string consists of alternating 1's and +'s, like 1 + 1 + 1 + 1 + 1. While it is being generated, the number of A's represents the nth Fibonacci number, the number of B's the (n+1)th Fibonacci number, and hence the total number of letters the (n+2)th Fibonacci number.

Another idea might be to have symbols that match more than one word -- like something called ADJECTIVE that matches any adjective. Or you could try to introduce something like a linguistic transformation; see the references.

A different interesting project is to "turn the grammar around" and use it to analyze strings of words, rather than create them. For instance, is a given simple sentence grammatical English? Or does a given picture of a face show teeth? A way to do that is to start with a string and the last rule of a grammar. You then try to find a match between things in your string and the second argument to the R command. (If the second argument contains choices, try every possibility.) If you find a match, substitute the first argument.

11. Acknowledgements

The ideas presented here are not particularly original. There is a large body of knowledge regarding grammars and parsing within computer science. Ken Kahn had the idea of using grammars with Logo. Work of Ira Goldstein and Mark Miller[7] has emphasized the grammatical aspects of Logo programming in general. The language SNOBOL[8] has been an influence. There have also been specific applications of grammars.[9,10,11,12,13] Finally, I must acknowledge the work of Seymour Papert and others in developing a new kind of learning environment based around Logo.

Thanks to Hal Abelson, Andy DiSessa, Ken Kahn, and Mark Miller for help with this paper.

References

1. The September-October 1976 issue contains four excellent articles of this kind; see also other issues.

2. Seymour Papert, "Uses of Technology to Enhance Education", MIT Artificial Intelligence Laboratory Logo Group Memo #8, June 1973.

3. Hal Abelson, Nat Goodman, and Lee Rudolph, "Logo Manual", MIT Logo Memo #7, June 1974; or contact the author for a draft of his manual in preparation.

4. Emmon Bach, Syntactic Theory, Holt, Rinehart, & Winston, 1973.

5. Andreas Koutsoudas, Writing Transformational Grammars: An Introduction, McGraw-Hill, 1966.

6. Barron Brainerd, Introduction to the Mathematics of Language Study, American Elsevier, 1971.

7. Ira Goldstein and Mark Miller, "Intelligent Tutoring Programs: A Proposal For Research", MIT

Logo Group Working Paper #50, 1976.

8. James F. Gimpel, Algorithms in SNOBOL4, Wiley, 1976.

9. William A. Woods, "Mathematical Linguistics and Automatic Translation", Harvard University Aiken Computation Laboratory, Report No. NSF-19 (September 1967).

10. David E. Rumelhart, "Notes on a Schema for Stories", in Representation and Understanding, Bobrow and Collins, ed., 1975, p.211.

11. Gahan Wilson, "The Science Fiction Horror Movie Pocket Computer", in National Lampoon: The Paperback Conspiracy, Warner, 1974.

12. Terry Winograd, "Linguistics and Computer Analysis of Tonal Harmony", Journal of Music Theory, 12:1 (1968), p.2.

13. A. C. Shaw, "A Formal Picture Description Scheme as a Basis For Picture Processing Systems", Information and Control, 14, 1969, p.5.

Appendix: a Logo implementation

```
TO SAY :PROCNAME
10 MAKE "STRING :PROCNAME
20 RUN :PROCNAME
30 PRINT :STRING
40 SAY :PROCNAME
END

TO R :SYMBOL :REPLACEMENT
10 MAKE "STRING MATCHREPLACE :STRING :SYMBOL
END

TO MATCHREPLACE :STRING :SYMBOL
10 IF (EMPTYP :STRING) OUTPUT :STRING
20 TEST ((F :SYMBOL) = (F :STRING))
30 IFTRUE OUTPUT SENTENCE (EXPANSION :REPLACEMENT) (MATCHREPLACE (BUTFIRST :STRING)
:SYMBOL)
40 IFFALSE OUTPUT SENTENCE (FIRST :STRING) (MATCHREPLACE (BUTFIRST :STRING) :SYMBOL)
END

TO EXPANSION :STRING
10 IF EMPTYP :STRING THEN OUTPUT []
20 TEST LISTP FIRST :STRING
30 IFFALSE OUTPUT SENTENCE (FIRST :STRING) (EXPANSION BUTFIRST :STRING)
40 IFTRUE OUTPUT SENTENCE (EXPANSION RANDCHOICE FIRST :STRING) (EXPANSION BUTFIRST
:STRING)
END

TO RANDCHOICE :LIST
10 OUTPUT ITEM (RANDOM 1 (COUNT :LIST)) :LIST
END
```

(RANDOM gives a random integer in the range X through Y; ITEM the Nth item of list L.)

Cyclic rule application:

```
TO CYCLETHRU :PROCNAME
10 MAKE "CHANGEFLAG "FALSE
20 RUN :PROCNAME
30 IF (:CHANGEFLAG = "TRUE) CYCLETHRU :PROCNAME
END
```

Change line 20 of SAY:

```
20 CYCLETHRU :PROCNAME
```

And change line 25 of MATCHREPLACE:

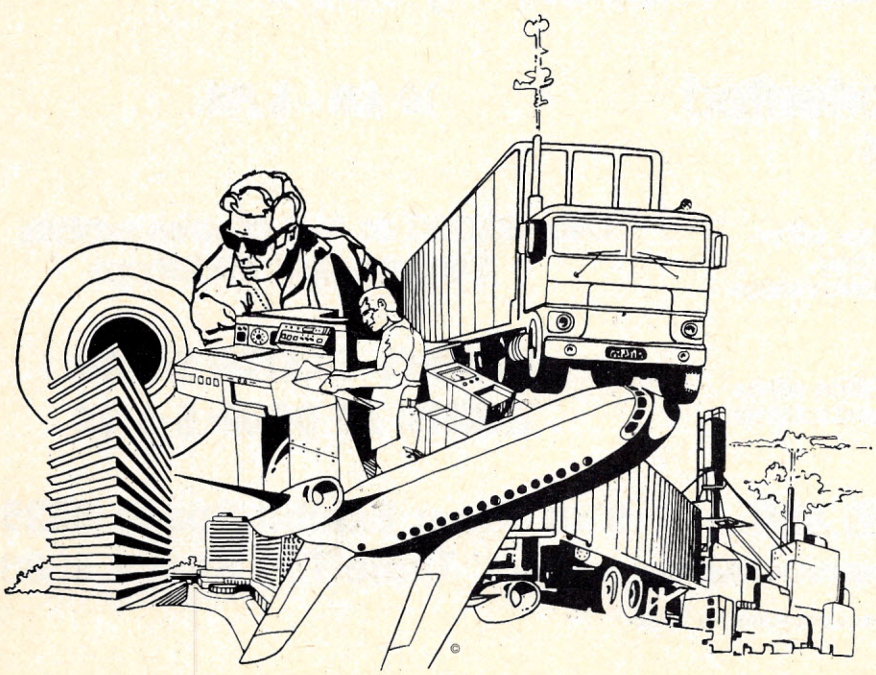
```
25 IFTRUE MAKE "CHANGEFLAG "TRUE
```

The R.ONCE feature:

```
TO R.ONCE :SYMBOL :REPLACEMENT
10 MAKE "REPLACEMENT (EXPANSION :REPLACEMENT)
20 MAKE "STRING MATCHREPLACE :STRING :SYMBOL
END
```

How to write a computer simulation

David H. Ahl



This article demonstrates how to write a computer simulation to be used for classroom instruction in some other subject. The object is not to teach about computers or computer programming; indeed very little programming knowledge is required to write a useful and effective simulation.

Nor is the purpose of this article to set down guidelines for determining instructional objectives or choosing subjects appropriate for simulation. The booklet *Designing Classroom Simulations* by Glenn Pate and Hugh Parker, Jr., fulfills these needs nicely. It should be noted, however, that not all subjects suitable for a classroom simulation or game are equally suitable for one on the computer.

For the most part, simulations and games have been found to be an extremely effective way of teaching a wide variety of subjects. Several studies have described the learning effectiveness of games (Allen, Allen, and Ross, 1970; Boochock and Schild, 1968; Fletcher, 1971). Learning games also generally create an intense and often enjoyable interpersonal experience. This is due in part to the interdependent task structure that requires interaction among the players (Inbar, 1968). The rationale for computer simulations is very compelling given the unique ability of such models to compress time, duplicate expensive, massive, delicate, or dangerous equipment, and produce large sample sizes (Braun, 1972).

However, lest we get overly ardent in using simulations, it should also be noted that certain things can be learned perfectly well by reading about them in a book, and to write a simulation is probably a waste of time to everyone but the writer, who will inevitably have learned more about the subject than when he or she started.

Presented at CCUC/5 (Conference on Computers in the Undergraduate Curricula) in Pullman, Washington, June 1974.

The Subject

In response to current events, the subject for our computer simulation is electrical energy generation and usage. The objectives of this simulation are:

1. To develop within the student the ability to recognize the need for advance planning and construction of electrical generation capacity.
2. To develop within the student the ability to plan an economically and environmentally sound electrical generation policy using five major fuel sources—coal, oil, gas, nuclear, and water.

Background Data and Assumptions

After poring over innumerable reference sources, you will quickly discover that:

1. There are far more facts and data available than you can use.
2. Several vital facts that you need are unavailable.

Here you can either quit before you've wasted too much time, or fearlessly plunge ahead. With a surge of optimism, we chose the latter course.

Ultimately, the following facts were obtained:

1. The demand for electricity is increasing 5.4% per year. In 1971, usage was 1.466 trillion kwh.
2. Some mathematical gyrations indicate that to produce 1.466 trillion kwh requires approximately 308 million kw of generation capacity plus a 20% margin of reserve (for extra-hot days, generator failures, emergencies, etc.) or a total capacity of 367 million kw.
3. The amount of generating capacity operational in 1971 is as follows:

Fuel	*Generating Capacity*	*Annual Growth*
Coal	167 million kw	6.7%
Fuel Oil	50	7.4
Gas	87	8.4
Nuclear	7	54.8
Hydro	56	5.2
Total	367	

4. Relationships between generation, fuels, and their supply is as shown in Table I.

FLOWCHART FOR ELECTRIC POWER GENERATION SIMULATION

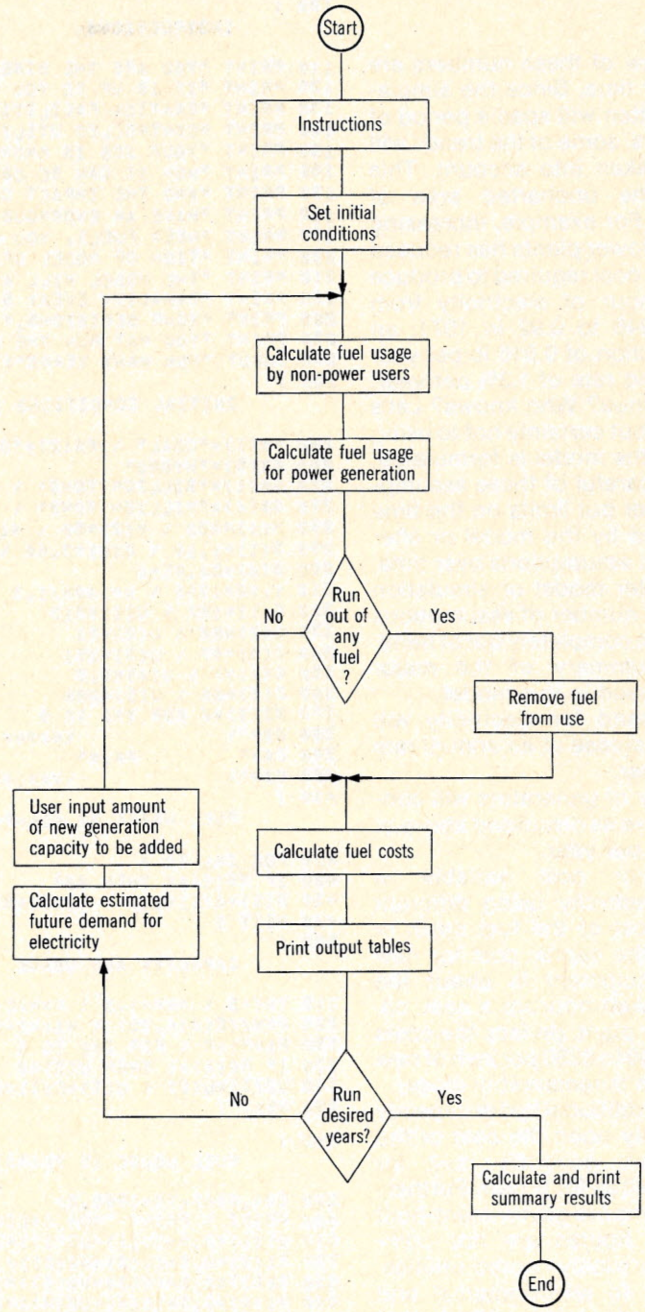


Figure 1

TABLE I

Fuel	*Amount to Generate 1 Kwh*	*Fuel Cost*	*Proven Reserves, United States*
Coal	.92 lb	$8.63/ton	830 billion tons
Fuel Oil	.00183 lb	$9.00/bbl	48 billion barrels
Gas	10.6 cu ft	$0.44/1000 cu ft	286 trillion cu ft
Nuclear	.0000075 lb	$530/lb	1.05 million tons
Hydro	n/a	$0.50/1000 kwh	190 million kw

Naturally, few of these numbers are constant over time. Since the simulation to be written will span a period of 50 to 100 years, some of the trends will have to be taken into account. This leads into the uncharted area of assumptions. For example, increasing efficiency of power plants has reduced the amount of coal required to produce one kilowatthour of electricity from 1.29 lb. in 1946 to 0.92 in 1971; an average reduction of 0.015 lb per year or a compound rate of 1.3% per year. Will this continue? Who knows? Let's assume it will but certainly not to using zero fuel. As the model is formulated, one must be careful of these assumptions and either put limits on the time period covered by the model or alter the trends and assumptions over time.

Any computer model or simulation requires a fair number of assumptions if it is not to get completely out of hand. Here is a summary of the major assumptions used in this model.

1. The demand for electricity will continue to increase at an annual rate of 5.4% per year.
2. Efficiency of generators will continue to improve as discussed above at a rate of 1.3% per year.
3. The major cost variable in generating electricity using different fuels is the cost of the fuel itself. In other words, the capital cost for new generating equipment is about the same per kilowatt hour for a coal, oil, gas, or nuclear plant. (In fact, the costs range from $240 to $280 per kwh of new capacity which is remarkably close).
4. Cost for additional hydro capacity depends greatly upon the cost of the dam which are generally built for irrigation as well as for power. Furthermore, most dams are government-built and the cost figures are not comparable with privately-owned utilities. However, due to environmental and other considerations, it is unlikely that new hydro capacity can be added at a rate of much over 5.5% per year. For the model, this rate is assumed.
5. Fuel reserves will not change significantly over time. This assumption is subject to the most debate. Discoveries of new natural gas sources, for example, have been keeping pace with usage. As fuel prices rise, new sources become economically feasible to exploit; deep coal mines or ones with thinner veins, for examples. Also, new recovery methods may be discovered or become cost effective; for example, the recovery of shale oil. In the computer program, it is wise to

```
10 PRINT "PROGRAM !POWER!"
20 PRINT "SIMULATION OF POWER GENERATION, FUEL REQUIREMENTS, AND COSTS"
30 PRINT "(VERSION 1-A, JAN 1974, WRITTEN BY DAVID H. AHL)"
40 INPUT "WANT INSTRUCTIONS (Y OR N)";A$ \ PRINT \ IF A$="N" THEN 230
100 !
       INSTRUCTIONS

110 PRINT "YOU ARE THE DIRECTOR OF THE U.S. FEDERAL POWER COMMISSION."
120 PRINT "IT IS UP TO YOU TO APPROVE THE BUILDING OF NEW POWER GEN-"
130 PRINT "ERATION FACILITIES FOR BOTH INVESTOR-OWNED AND GOVERNMENT"
140 PRINT "CONTROLLED UTILITIES.  THEREFORE, AN IMPORTANT PART OF"
150 PRINT "YOUR JOB IS KNOWING HOW MUCH ADDITIONAL POWER IS REQUIRED,"
160 PRINT "HOW IT CAN BE OBTAINED (I.E., USING WHAT TYPE OF FUEL),"
170 PRINT "AND THE IMPACT ON FUEL RESERVES.  SINCE YOUR ACTIONS WILL"
180 PRINT "HAVE AN ENORMOUS IMPACT ON FUTURE GENERATIONS (WHO ALSO"
190 PRINT "NEED FUEL), YOU HAVE AT YOUR DISPOSAL A COMPUTER SIMULA-"
200 PRINT "TION OR MODEL OF THE U.S. POWER GENERATION SYSTEM."\PRINT
210 PRINT "THE MODEL WILL ASK FOR YOUR DECISIONS ON NEW GENERATING"
220 PRINT "CAPACITY EVERY 5 YEARS AND THEN GIVE YOU THE RESULTS OF"
225 PRINT "YOUR DECISIONS." \ PRINT
230 PRINT "YOU MAY RUN THE MODEL FOR 25 TO 100 YEARS."
240 INPUT "HOW MANY YEARS";Y3 \ IF Y3<25 OR Y3>100 THEN 230
250 !
       INITIAL CONDITIONS AND VALUES

260 F$(1)="COAL" \ F$(2)="OIL" \ F$(3)="GAS" \ F$(4)="NUCLEAR"
265 F$(5)="HYDRO"
270 G$(1)="BILLION TONS" \ G$(2)="BILLION BARRELS" \ G$(3)="TRILLION CU FT"
275 G$(4)="MILLION TONS" \ G$(5)="MILLION KW"
280 H(1)=832 \ H(2)=50 \ H(3)=700 \ H(4)=1.05 \ H(5)=190  ! 1966 RESERVES
290 S(1)=1.16 \ S(2)=3.66 \ S(3)=45
295 S(4)=34.8E-6                              ! OTHER FUEL USAGE
310 Y,Y2=1966 \ W=1466/1.3 \ C(6)=367   ! YEAR, USAGE, GEN CAPACITY
320 C(1)=167 \ U(1)=613          ! COAL CURRENT CAPACITY, 1966 USAGE
330 C(2)=50 \ U(2)=76
340 C(3)=87 \ U(3)=251
350 C(4)=7 \ U(4)=5.5
360 C(5)=56 \ U(5)=200
370 R2(I)=0 FOR I=1 TO 5          ! USAGE OF RESERVES
380 B$="\            \#####          ######"
390 C$="          ##.#"
400 D$="\            \###.##         #######        ###.##"
410 !
       FUEL USAGE BY NON-POWER GENERATION SOURCES

420 FOR I=1 TO 4
425 IF R9(I)=1 THEN 440
430 S(I)=S(I)*1.3 \ R2(I)=R2(I)+S(I)
440 NEXT I
500 !
       CAPACITY AND USAGE

510 Y=Y+5 \ W=W*1.3 \ K=W/C(6)   ! YEAR, DEMAND, CONSTANT
520 P=W/(C(6)*.05) \ W1=W1+5.75*W        ! % FUEL USAGE, CUM GENERATION
530 P1=P1+P \ FOR I=1 TO 5
535 IF R9(I)=1 THEN U(I)=0
540 U2(I)=U(I) \ U(I)=C(I)*K             ! ALLOCATE GENERATION BY FUEL
550 NEXT I

560 !
       FUEL USAGE (5 YEARS)

590 E9=.9867^(Y-1968.5)                  ! 5-YEAR EFFICIENCY FACTOR
600 PRINT \ PRINT "NOW CALCULATING 5-YEAR FUEL USAGE" \ PRINT
610 R1(1)=2.5*(U(1)+U2(1))*.92*E9/2000  ! COAL BILLION TONS
620 R1(2)=2.5*(U(2)+U2(2))*.00183*E9             ! OIL BILLION BBL
630 R1(3)=2.5*(U(3)+U2(3))*10.6*E9/1000          ! GAS TRILLION CU FT
640 R1(4)=2.5*(U(4)+U2(4))*3.75                  ! NUCLEAR TONS
650 FOR I=1 TO 3
655 IF R9(I)=1 THEN 710
660 R2(I)=R2(I)+R1(I)
670 H1(I)=(H(I)-R2(I))*100/H(I)          ! % CUM FUEL USAGE
680 IF H1(I)>10 THEN 710
690 PRINT "THE U.S. HAS VIRTUALLY RUN OUT OF "F$(I)" !" \ R9(I)=1
710 NEXT I
715 IF R9(4)=1 THEN 760
720 R2(4)=R2(4)+R1(4)*1E-6
730 H1(4)=(H(4)-R2(4))*100/H(4)          ! % CUM FUEL USAGE
740 IF H1(4)>10 THEN 760
750 PRINT "THE U.S. HAS VIRTUALLY RUN OUT OF NUCLEAR FUEL !" \ R9(4)=1
760 R(5)=C(5)*100/H(5)
770 IF R(5)>=100 THEN PRINT "YOU HAVE NO REMAINING HYDRO CAPACITY !"
780 FOR I=1 TO 4
785 IF R7(I)=1 THEN 810
790 IF R9(I)=0 THEN 810
800 C(6)=C(6)-C(I) \ C(I)=0 \ R7(I)=1
810 NEXT I
900 !
       COST OF FUEL PER KW AND TOTAL
```

```
910 E9=.9867^(Y-Y2)*1000
920 D(1)=.92*E9*8.63/2000                 ! COST=FUEL/KW * COST/QUAN
930 D(2)=.00183*E9*9
940 D(3)=.0106*E9*.44
950 D(4)=7.5E-6*E9*530
960 D(5)=.5
970 T(6)=0
980 FOR I=1 TO 5
990 T(I)=U(I)*D(I) \ T(6)=T(6)+T(I)      ! TOTAL FUEL COSTS
1000 NEXT I
1010 T1=T1+T(6) \ D(6)=T(6)/W
1290 !
      5-YEAR RESULTS

1300 PRINT \ PRINT "RESULTS FOR"Y \ PRINT
1310 PRINT "","CAPACITY","USAGE","USAGE (%"
1320 PRINT "FUEL","(MILLION KW)","(BILLION KWH)","OF CAPACITY)"
1330 FOR I=1 TO 5 \ IF R8(I)=1 THEN 1350
1340 PRINT USING B$,F$(I),C(I),U(I)
1350 NEXT I
1360 PRINT USING B$+C$,"TOTAL",C(6),W,P
1361 IF P>80 THEN 1362 ELSE IF P<70 THEN 1366 ELSE GOTO 1370
1362 PRINT \ PRINT "YOUR MARGIN OF RESERVE CAPACITY IS ONLY"INT(100-P)"%"
1363 IF P>90 THEN 1364 ELSE PRINT "LEADING TO NUMEROUS BROWNOUTS."\GOTO 1368
1364 PRINT "LEADING TO MAJOR PROBLEMS IN INDUSTRIAL PRODUCTION AND URBAN"
1365 PRINT "TRANSPORTATION SYSTEMS." \ GOTO 1368
1366 PRINT "YOUR RESERVE CAPACITY MARGIN OF"INT(100-P)"% IS TOO GREAT."
1367 PRINT "THE INVESTMENT FOR NEW PLANTS WAS NEEDED MORE IN OTHER AREAS."
1368 PRINT "YOUR SUCCESSOR IS ABOUT TO BE APPOINTED UNLESS THERE IS A"
1369 PRINT "DRAMATIC IMPROVEMENT IN THE NEXT 5 YEARS."
1370 PRINT \ PRINT "","FUEL COST","TOTAL FUEL $","REMAINING FUEL"
1380 PRINT "FUEL","PER 1000 KWH","(MILLION $)","(% OF 1966)"
1385 H1(5)=(H(5)-C(5))*100/H(5)
1390 FOR I=1 TO 5 \ IF R8(I)=1 THEN 1420
1400 PRINT USING D$,F$(I),D(I),T(I),H1(I)
1420 NEXT I
1430 PRINT USING D$,"TOTAL",D(6),T(6)
1440 IF Y-Y2>=Y3 THEN 1700
1450 PRINT \ PRINT "EST. DEMAND FOR ELECTRICITY IN "Y+5"IS"W*1.3"BILLION KWH"
1460 PRINT "GENERATING CAPACITY IN"Y"IS"C(6)"BILLION KW."
1470 PRINT "DESIRABLE GENERATING CAPACITY IN"Y+5"IS"W*.325"BILLION KW."
1480 PRINT
1490 R8(I)=R9(I) FOR I=1 TO 4
1500 !
      INPUT NEW VALUES

1510 PRINT "AMOUNT OF NEW CAPACITY TO BE ADDED IN NEXT 5 YEARS (BILLION KW)"
1520 C1(6)=0
1530 FOR I=1 TO 5 \ IF R9(I)=1 THEN 1560
1540 PRINT F$(I), \ INPUT N(I)
1542 IF I<>5 THEN 1550
1543 IF N(5)<=.3*C(5) THEN 1544 ELSE A=.3*C(5) \ GOTO 1546
1544 IF C(5)+N(5)<190 THEN 1550 ELSE A=190-C(5)
1546 PRINT "THAT'S TOO MUCH ADDITIONAL HYDRO CAPACITY." A"IS THE LIMIT."
1548 GOTO 1540
1550 C1(I)=C(I)+N(I) \ C1(6)=C1(6)+C1(I)
1560 NEXT I
1570 IF C1(6)>W*.26 THEN 1600
1580 PRINT "THAT'S NOT ENOUGH ADDITIONAL CAPACITY TO MEET THE DEMAND."
1590 PRINT "YOU NEED A MINIMUM OF"INT(W*.26-C1(6))"BILLION KW OVER"
1595 PRINT "THE CURRENT CAPACITY OF"C(6)".  PLEASE TRY AGAIN." \ GOTO 1510
1600 !
      UPDATE 5 YEAR AGO FIGURES

1610 FOR I=1 TO 6
1620 C2(I)=C(I) \ C(I)=C1(I)
1630 NEXT I
1640 GOTO 500
1700 !
      END SUMMARY

1710 PRINT \ PRINT \ PRINT "SUMMARY RESULTS -- 1966 TO"Y \ PRINT
1715 Y1=Y-1966 \ H1(5)=H(5)-C(5)         ! TOTAL YEARS, % HYDRO USAGE
1720 PRINT "TOTAL ELECTRICITY GENERATED ="INT(W1)"BILLION KWH."
1725 PRINT "AVERAGE PERCENT OF CAPACITY IN USE ="P1*5/Y1 \ PRINT
1730 Y1=Y-1966 \ PRINT Y1"YEAR FUEL COST SUMMARY:"
1740 PRINT "      TOTAL COST = $"5*T1"MILLION"
1745 PRINT "      AVERAGE COST PER YEAR = $"5*T1/Y1
1750 PRINT "      AVERAGE COST PER 1000 KWH = $"5*T1/W1
1760 PRINT \ PRINT "","RESERVES","RESERVES"
1770 PRINT "FUEL","IN 1966","IN"Y \ PRINT
1790 FOR I=1 TO 5
1795 A=H(I)-R2(I) \ IF A<0 THEN A=0
1800 PRINT F$(I),H(I),H(I)-R2(I),G$(I)
1810 NEXT I
1820 PRINT \ PRINT \ PRINT "FUEL","% REMAINING OF 1966 RESERVE" \ PRINT
1830 FOR I=1 TO 5
1835 IF H1(I)<0 THEN H1(I)=0
1840 PRINT F$(I),H1(I)
1850 NEXT I
2000 END
```

leave all these factors as variables so that you, or an enterprising student, can try out other alternatives or "improve" the simulation.

6. The model will assume a self-sufficient energy policy; that is, only fuel reserves within the United States can be called upon. If you're an optimistic internationalist, you can easily change this assumption in the one line (280) where these values are set. Again, the general principle is: leave everything a variable.

7. All available fuel cannot be used for the generation of electricity. We've assumed the following percentages are available: coal 55%, oil 25%, gas 25%, nuclear 90%.

You'll discover other assumptions that have to be made or facts that have to be gathered as you actually write the model but if you have most of it beforehand, you'll have an easier time.

Writing the Model

The temptation, of course, is to plunge into the computer program. DON'T! Write a flowchart first. Virtually all interactive models and simulations for teaching follow these steps, exclusive of instructions.

1. Set initial conditions.
2. Perform the calculations for one period, cycle, steady-state at the start, etc. Output this.
3. Accept input from the user.
4. Perform calculations for the next period using inputted data. Output this.
5. Go back to Step 3 until the model has run the desired number of steps, periods, etc. At this point go to Step 6.
6. Calculate and output final results.

These steps have been transformed into the simplified flowchart shown in Figure 1.

It's probably also a good idea to state any formulas used in the model. Other than percentage calculations, the electrical generation model didn't use many formulas. Here are the most notable ones:

Five year fuel usage = 5x (kw Generation Year 1 + kw Generation Year 5)÷ 2x Efficiency improvement factor x fuel per kw.

Cost of fuel per yr. = kw Generation x Efficiency improvement factor x fuel per kw x cost per unit of fuel

It's also worthwhile to state the format of the output and assign letter variables right on the mocked-up output. For example:

Fuel	Capacity (Million kw)	Usage (Billion k wh)	Usage (% of Capacity)
F$(1)	c(1)	u(1)	p
F$(2)	c(2)	u(2)	p
etc.			

Now, finally, you're ready to write the program. If you've done the previous steps thoroughly, the writing of the program will be trivial (well, perhaps not trivial, but at least straightforward). It's probably good practice to write the program in sections. Worry about the calculations section first. If you can hack through that, the other sections will fall nicely into place. While you're doing the calculations section, jot down the variables that will have to be set in other sections like initial conditions or user input.

Use comments liberally throughout your program. You'll be glad you took the time when you start to debug the program or even look at it a few weeks later.

Add some elements to keep the program interesting to the user. If a value gets out of control, don't just print a message but try to describe the impact of that value. For example, instead of printing:

RESERVE POWER MARGIN = 12%

the user will learn more and be more interested if you print something like this:

YOUR MARGIN OF RESERVE POWER IS ONLY 12%. THERE HAVE BEEN NUMEROUS BROWNOUTS AND YOUR SUCCESSOR IS ABOUT TO BE APPOINTED UNLESS THERE IS A DRASTIC IMPROVEMENT.

Look over the power-generation program. Not for the specifics of how it works, but rather note the liberal use of comments, its division into sections, and explanations to the user.

Writing The User's Guide

Most simulations are of sufficient complexity that all the necessary background information cannot be included in the computer instructions in the program. Even with the simplest simulations, additional background information and suggested readings for further pursuit of the topic are desirable for the student to have in hand. The user's guide need not be long but it should include the following:

A. *User Section*

1. A brief description of the event, system, or apparatus being simulated.
2. Instructions for running the program and a sample run.
3. Questions, exercises, and/or activities to do with the program output. These should be as open-ended as possible, consistent with the student fulfilling your learning objectives.

```
RUNNH
PROGRAM 'POWER'
SIMULATION OF POWER GENERATION, FUEL REQUIREMENTS, AND COSTS
(VERSION 1-A, JAN 1974, WRITTEN BY DAVID H. AHL)
WANT INSTRUCTIONS (Y OR N)? Y

YOU ARE THE DIRECTOR OF THE U.S. FEDERAL POWER COMMISSION.
IT IS UP TO YOU TO APPROVE THE BUILDING OF NEW POWER GEN-
ERATION FACILITIES FOR BOTH INVESTOR-OWNED AND GOVERNMENT
CONTROLLED UTILITIES.  THEREFORE, AN IMPORTANT PART OF
YOUR JOB IS KNOWING HOW MUCH ADDITIONAL POWER IS REQUIRED,
HOW IT CAN BE OBTAINED (I.E., USING WHAT TYPE OF FUEL),
AND THE IMPACT ON FUEL RESERVES.  SINCE YOUR ACTIONS WILL
HAVE AN ENORMOUS IMPACT ON FUTURE GENERATIONS (WHO ALSO
NEED FUEL), YOU HAVE AT YOUR DISPOSAL A COMPUTER SIMULA-
TION OR MODEL OF THE U.S. POWER GENERATION SYSTEM.

THE MODEL WILL ASK FOR YOUR DECISIONS ON NEW GENERATING
CAPACITY EVERY 5 YEARS AND THEN GIVE YOU THE RESULTS OF
YOUR DECISIONS.

YOU MAY RUN THE MODEL FOR 25 TO 100 YEARS.
HOW MANY YEARS? 50

NOW CALCULATING 5-YEAR FUEL USAGE


RESULTS FOR 1971

            CAPACITY         USAGE            USAGE (%
FUEL        (MILLION KW)     (BILLION KWH)    OF CAPACITY)
COAL          167              667
OIL            50              200
GAS            87              348
NUCLEAR         7               28
HYDRO          56              224
TOTAL         367             1466             79.9

            FUEL COST        TOTAL FUEL $     REMAINING FUEL
FUEL        PER 1000 KWH     (MILLION $)      (% OF 1966)
COAL          3.71            2477              99.65
OIL          15.40            3077              88.04
GAS           4.36            1516              89.45
NUCLEAR       3.72             104              99.97
HYDRO         0.50             112              70.53
TOTAL         4.97            7285

EST. DEMAND FOR ELECTRICITY IN  1976 IS 1905.8 BILLION KWH
GENERATING CAPACITY IN 1971 IS 367 BILLION KW.
DESIRABLE GENERATING CAPACITY IN 1976 IS 476.45 BILLION KW.

AMOUNT OF NEW CAPACITY TO BE ADDED IN NEXT 5 YEARS (BILLION KW)
COAL          ? 50
OIL           ? 0
GAS           ? 0
NUCLEAR       ? 50
HYDRO         ? 10

NOW CALCULATING 5-YEAR FUEL USAGE
RESULTS FOR 1976

            CAPACITY         USAGE            USAGE (%
FUEL        (MILLION KW)     (BILLION KWH)    OF CAPACITY)
COAL          217              867
OIL            50              200
GAS            87              348
NUCLEAR        57              228
HYDRO          66              264
TOTAL         477             1906             79.9

            FUEL COST        TOTAL FUEL $     REMAINING FUEL
FUEL        PER 1000 KWH     (MILLION $)      (% OF 1966)
COAL          3.47            3011              99.46
OIL          14.41            2878              84.74
GAS           4.08            1418              87.07
NUCLEAR       3.48             792              99.74
HYDRO         0.50             132              65.26
TOTAL         4.32            8230

EST. DEMAND FOR ELECTRICITY IN  1981 IS 2477.54 BILLION KWH
GENERATING CAPACITY IN 1976 IS 477 BILLION KW.
DESIRABLE GENERATING CAPACITY IN 1981 IS 619.385 BILLION KW.

AMOUNT OF NEW CAPACITY TO BE ADDED IN NEXT 5 YEARS (BILLION KW)
COAL          ? 60
OIL           ? 0
```

```
GAS            ? 0
NUCLEAR        ? 50
HYDRO          ? 15

NOW CALCULATING 5-YEAR FUEL USAGE


RESULTS FOR 1981

               CAPACITY        USAGE           USAGE (%
FUEL           (MILLION KW)    (BILLION KWH)   OF CAPACITY)
COAL             277             1140
OIL               50              206
GAS               87              358
NUCLEAR          107              440
HYDRO             81              333
TOTAL            602             2478           82.3

YOUR MARGIN OF RESERVE CAPACITY IS ONLY 17 %
LEADING TO NUMEROUS BROWNOUTS.
YOUR SUCCESSOR IS ABOUT TO BE APPOINTED UNLESS THERE IS A
DRAMATIC IMPROVEMENT IN THE NEXT 5 YEARS.

               FUEL COST       TOTAL FUEL $    REMAINING FUEL
FUEL           PER 1000 KWH    (MILLION $)     (% OF 1966)
COAL             3.25            3702            99.22
OIL             13.47            2772            81.60
GAS              3.82            1366            84.81
NUCLEAR          3.25            1432            99.14
HYDRO            0.50             167            57.37
TOTAL            3.81            9439

EST. DEMAND FOR ELECTRICITY IN  1986 IS 3220.8 BILLION KWH
GENERATING CAPACITY IN 1981 IS 602 BILLION KW.
DESIRABLE GENERATING CAPACITY IN 1986 IS 805.201 BILLION KW.


(Output for Period 1986-2011 is not shown)


RESULTS FOR 2016

               CAPACITY        USAGE           USAGE (%
FUEL           (MILLION KW)    (BILLION KWH)   OF CAPACITY)
COAL            3274            13095
OIL               50              200
GAS               87              348
NUCLEAR          287             1148
HYDRO            189              756
TOTAL           3887            15546           80.0

               FUEL COST       TOTAL FUEL $    REMAINING FUEL
FUEL           PER 1000 KWH    (MILLION $)     (% OF 1966)
COAL             2.03           26614            93.44
OIL              8.43            1686            64.80
GAS              2.39             831            72.72
NUCLEAR          2.04            2336            86.04
HYDRO            0.50             378             0.53
TOTAL            2.05           31845

SUMMARY RESULTS -- 1966 TO 2016

TOTAL ELECTRICITY GENERATED = 359261 BILLION KWH.
AVERAGE PERCENT OF CAPACITY IN USE = 80.3055

 50 YEAR FUEL COST SUMMARY:
     TOTAL COST = $ 809391 MILLION
     AVERAGE COST PER YEAR = $ 16187.8
     AVERAGE COST PER 1000 KWH = $ 2.25293

               RESERVES        RESERVES
FUEL           IN 1966         IN 2016

COAL            832             777.45          BILLION TONS
OIL             50              32.3996         BILLION BARRELS
GAS             700             509.022         TRILLION CU FT
NUCLEAR         1.05            .903427         MILLION TONS
HYDRO           190             190             MILLION KW


FUEL           % REMAINING OF 1966 RESERVE

COAL            93.4436
OIL             64.7993
GAS             72.7175
NUCLEAR         86.0407
HYDRO           1

READY
```

B. *Description Section*

1. Reprints or summaries of articles or information about the vent being simulated.
2. Background data and its source.
3. Key assumptions and their rationale.
4. Short bibliography for further reference.

C. *Computer Section* (optional)

1. Program listing.
2. Flowchart.

References — Computer Simulations

1. Ahl, D.H., (Ed.), *101 BASIC Computer Games,* Maynard, MA: Digital Equipment Corporation, 1973.
2. Allen, L.E., Allen, R.W., & Ross, Jr., "The Virtues of Nonsimulation Games." *Simulation and Games,* 1970, 1, 319-326.
3. Boocock, S.S., & Schild, E.O. (Eds.), *Simulation Games in Learning.* Beverly Hills, CA: Sage, 1968.
4. Braun, L., "Digital Simulation in Education." *J. Educational Technology Systems,* 1972, 1, 1, 5-27.
5. DeVries, D.S. & Edwards, K.J., "Learning Games and Student Teams: Their Effects on Classroom Process." *American Education Research Journal,* 1973, 10, 307-318.
6. Edwards, K.J., DeVries, D.L., & Snyder, J.P., "Games and Teams: A Winning Combination." *Simulation and Games,* 1972, 3, 247-269.
7. Fletcher, J.L., "The Effectiveness of Simulation Games as Learning Environments." *Simulation and Games,* 1971, 2, 425-454.
8. Inbar, M., "Individual and Group Effects on Enjoyment and Learning in a Game Simulating a Community Disaster." In S.S. Boocock & E.O. Schild (Eds.), *Simulation Games in Learning.* Beverly Hills, CA: Sage, 1968, pp. 169-190.
9. Pate, G.S. & Parker, H.A., Jr., *Designing Classroom Simulations,* Belmont, CA: Fearon, 1973.
10. Spencer, D.D., *Game Playing With Computers.* New York: Spartan Books, 1968.

References — Electric Power

1. "Power Industry Statistics for 1971." *Electrical World,* Sept. 15, 1972.
2. *Questions and Answers About the Electric Utility Industry,* Edison Electric Institute, 1973. ■

Computer Education at Lawrence Hall

by Robert Kahn

Lawrence Hall of Science at the University of California – Berkeley is a center in science education for the general public, for young children, for everyone. We're developing curriculum; we have a traveling van going around the northern Bay Area taking educational materials to teachers; we have a planetarium; we have public science education; we have an enrichment center, and we have computer activities.

Our computer activities are now far and away the largest single public-access project at the Lawrence Hall of Science. We have a time-sharing service which started out with one small Nova; now we have the capacity to handle approximately 100 terminals.

The heart of our system is a Data General NOVA modified by a company named Decision, in Berkeley, with a capacity of 60 simultaneous uses. We also have a Hewlett-Packard 2000B and we have the first production model of the Hewlett-Packard 3000.

Almost everything we're doing is in BASIC and in our version of PILOT, which we call NYLON. One of our major projects is collecting educational programs and putting them together in a documented format. We have a whole book describing the programs that we have available in our program library, and we're continuing to collect new ones.

We have thirty outside users on our system, mostly schools. We are running an educational time-sharing service where the costs are very low in terms of general time-sharing costs.

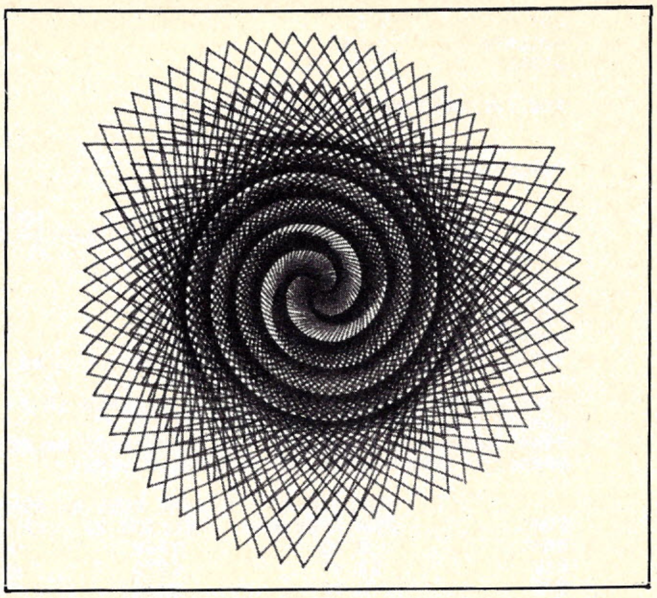
Here are some of the things that are going on at the Lawrence Hall of Science, to give you a flavor of all the kinds of computer activities there.

The Computer Education Project is run by students, like myself, from the Berkeley campus. These students are both graduates and undergraduates and they teach classes to children, run workshops, develop course materials, and do other related things.

The computer project at the Hall has three objectives. The first objective is to educate adults and children about computers. We are following in the mold as Darmouth but on a very much smaller scale. The idea is to introduce people to the computer as a tool, and allow them to do what they want to do on it. We want them to know what computers can do and how people might be able to use them. We're not quite so concerned with presenting curriculum via the computer for use in instruction. The second goal is to offer what we call "low-cost computing" to the public. We want to provide a place where people can come and use the computer, while keeping the cost down. We have a policy that when we're not using our computer rooms, we open them to the general public for 75¢ an hour—what we call public computing time. It's like going to the movies. You buy a ticket at the front desk when you enter the Hall, and you spend your hour playing games, writing programs, drawing pictures, or whatever you want. This goes for anybody, kids or adults, college students from Berkeley, professors, anybody that wants to. But we cannot afford to provide long-term storage with this policy. The public account has a small amount of disk space available, but we wipe this space clean each week.

We have a school visitation program where about 400 Bay Area children come to the Lawrence Hall of Science every week, all year long. The program is booked for a year ahead. It's a field-trip program with five activities. The computer is one. The computer activity consists of a set of six games, purposely selected to be diverse in strategy, concept, and difficulty. As many as three children share each terminal. A monitor program asks the children for their first names and then proceeds to rotate the game going from child to child by telling each child when it is his or her turn and asking what game he/she wishes to play. This program also keeps a record of the age and sex of each child, of which games the child chose to play, and for how long. These data help us to find out which games are most popular with students of a given age and sex, and we can then make necessary changes and modifications in the program.

In the past year we have hosted several special activities. For example, we hosted a Bay Area Computer Fair in which junior and senior high school students from all



Lawrence Hall of Science presents

COMPUTER EXPO

Contests

ENTER OUR CONTESTS AND WIN MONSTER POSTERS, BOOKS, SUBSCRIPTIONS, AND LHS MEMBERSHIPS. CONTESTS WILL BE HELD IN PROGRAMMING, LOGIC GAMES, TAPE WINDING, AND MAZES

around the San Francisco Bay Area came up to the Hall to play games and write programs, for three days in September. Many vendors were also present at the Fair, displaying their latest advances in hardware.

The Hall of Science has a public area (like a science museum) on its topmost level. We have placed four CRT terminals there that are free to use for anybody who has already been admitted to the Hall.

We've pre-programmed these CRT terminals with a variety of computer activities. One CRT had the ELIZA Program -- one of our staff re-wrote Weizenbaum's ELIZA Program to run on our Decision Computer, and this program has been a favorite with the public. During the past two months, we have temporarily replaced ELIZA with a program that simulates the path of a comet. You control various input parameters such as initial position, speed, and distance from the sun, and the computer plots the path of your comet.

Two other CRT terminals are programmed to play our most popular games. The fourth one controls a tone box that we constructed. The tone box consists of a set of twelve oscillators, on which you can -- similar to the Dartmouth music generator -- type in a set of notes and the computer will play back your music. We have several programs to allow you to compose music, and if you compose a song that you really like, we'll put it in the library of songs. Note that these activities are all free to the general public, once they are admitted to the Hall of Science.

Films

A VARIETY OF FILMS WILL BE SHOWN THROUGHOUT THE FOUR-DAY FAIR INCLUDING:

COMPUTERS: CHALLENGING MAN'S SUPREMACY

COMPUTER ANIMATION

COMPUTERS AND HUMAN BEHAVIOR

RIGHT OF PRIVACY

ROBOTICS: ISAAC ASIMOV'S (CLANK! CLANK!) ARTIFICIAL MAN and many more . . .

In the afternoons and on Saturdays we offer courses about computers for children and adults. We developed a course for younger children, eight, nine, and ten years old, called "Creative Play with the Computer." It is an introductory course in which we encourage them to play games on the teletypes, use our Tektronix 611 storage scope (driven by a NOVA), our Hewlett-Packard 7200A plotter, driven in BASIC by any of the computers, and the tone box. The aim is to give them a general introduction to the kinds of things that computers can do. Some of the children are sharp enough to go on to doing a little bit of PILOT programming at the end of this course. Many come back for other courses.

None of these projects could function if it weren't for the fact that we have a committed and dedicated staff from Computer Sciences and other departments who work half time and often put in more time on their own.

We also have a course that we call "Computer Theory," for children who really want to know more about computers than programming, who want to know how the computer works. This quarter, we're also offering a course for adults in the Berkeley area. Because their children were taking courses, the parents became interested, so we thought we could also do something for adults.

General Public

Unlimited computer time

Demonstrations on homemade computers, voice synthesizers, and calculators

The adult course is structured along the lines of Creative Play with the Computer. In many cases, adults are much more afraid of the machine than are children. Many adults have misconceptions about computers which the course attempts to dispell while providing a positive, hands-on experience.

During the past year, another thing has been happening: We've been getting calls -- as many as about three a day -- from schools around the Bay Area, as they learn about our programs. Often it's a matter of a teacher wanting to bring a class of thirty students for one visit to our computer lab. But several teachers have been interested in starting a program where they can bring their class to the Hall once a week for one or more eight-week sessions. In these classes, the Hall becomes an extended classroom for the schools. The word is out that we teach BASIC programming, and many of the teachers want to integrate programming into their curriculum.

These classes work out very well because we have a group of students that is pretty much homogeneous, coming from one school (unlike the classes on Saturday) to the Lawrence Hall of Science. One of our staff meets with the teacher each week to plan the week's lesson, and the students take what they learn back to the classroom. So we achieve a dialog going back and forth between the Hall and the classroom. We now have twelve of these classes, which is as many as we can handle.

Finally, we are now providing, as I said, the beginning of a campus time-sharing service, and little by little we're starting to hear from various departments around the Berkeley campus. The Biology 1 program is one example. For that course, we provide computer time, teacher training, and programs. Our staff spent three Monday nights training the TAs in the Biology 1 course to get them ready for the 600 students who were to use simulation programs as part of their coursework in Biology 1. We also provide the programming expertise needed to get all of the simulation programs checked out and running. For a Division course taught by Charlie Bass in the Computer Sciences they merely rented terminals from us. Also, Charles Woodson has offered a course in the School of Education on computer-based instruction, using our computers. He has two terminals in the School of Education and expects to get more.

So, little by little, we have been backing into becoming the campus time-sharing service. There were some campus problems with this situation at first, but these problems are being worked out, and more and more people are starting to call from the campus. So, we are promoting the type of computer service that Dartmouth has -- only on a smaller scale. We have a similar setup, using the same language. We're riding a tide of high activity at this point. Things are going so fast that we haven't had time to stop, evaluate, and look ahead. We are just trying to keep up with the growing demand for our service. ■

A New Fast Sorting Algorithm

How to sort extremely fast with a minimum of comparisons, and a minimum of programming steps between comparisons.

Richard Hart*

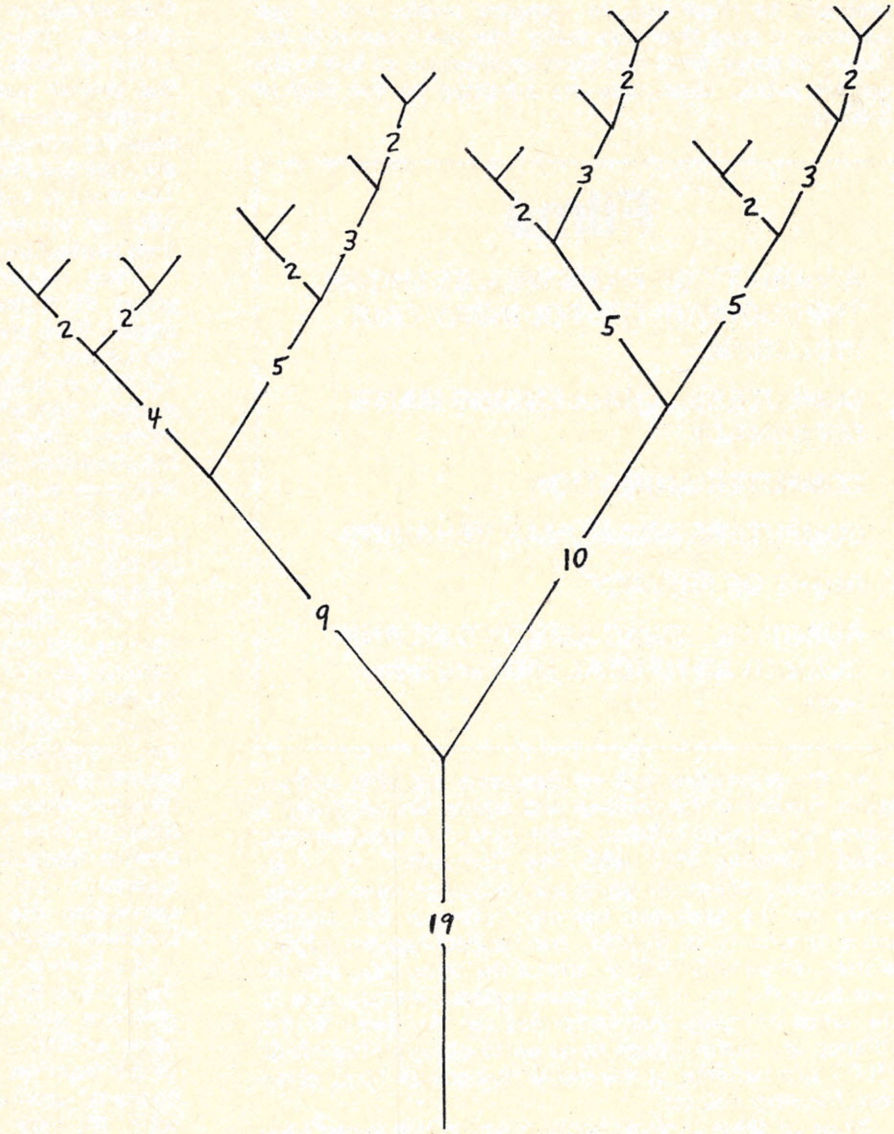


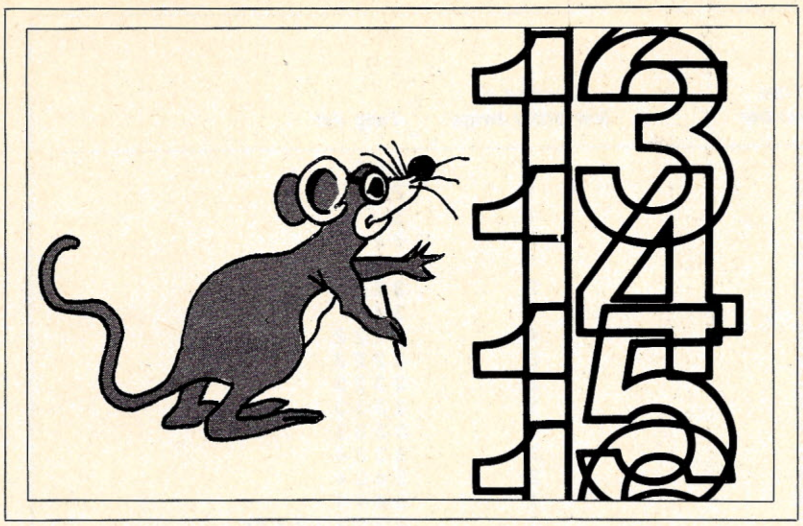
Fig. 1. Tree number 19, with 19 leaves and nearly equal branches.

The speed or efficiency of an internal sorting algorithm is directly related to the number and complexity of programming steps which are actually executed during its operation. Because all the items being sorted must be compared to one another to place them in order, programmers have realized that by making certain intelligent comparisons, the total number of comparisons executed in an algorithm may be reduced to a theoretical minimum.

This article starts with the theory described by Luther J. Woodrum in the *IBM Systems Journal* and describes an algorithm that not only uses a minimal number of comparisons, but also executes a minimal number of programming steps between each comparison.

The algorithm was written to use as few comparisons as possible, to have as few steps between each comparison as possible, to take advantage of natural sequencing, to preserve the order of equals (or even the reverse order of equals), to avoid moving records around, to use as little memory as possible (one working array), and to be a modular, easily understood program written in BASIC.

29 Concord Ave., Apt. 609, Cambridge, MA 02138



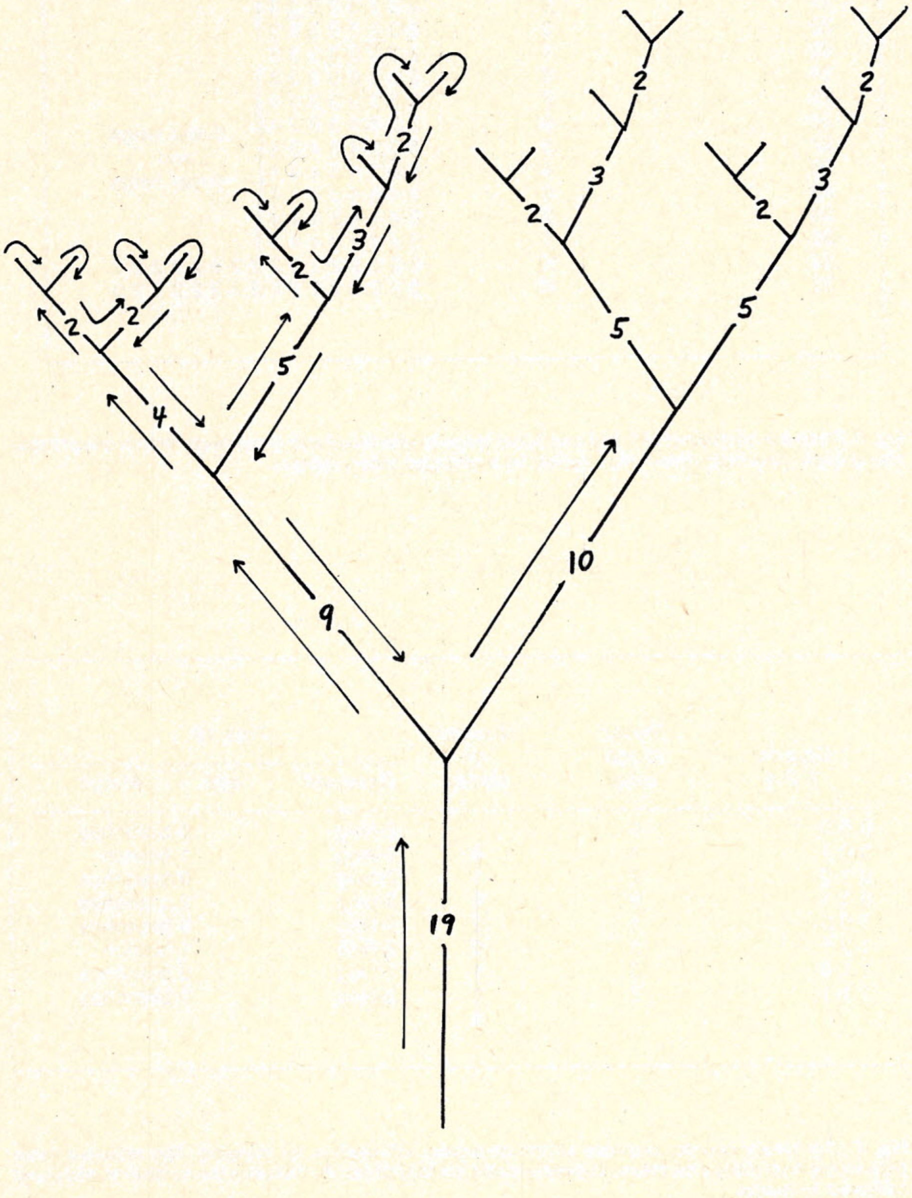


Fig. 2. Woodrum's algorithm for sorting N records with minimal comparing: follow the tree structure and merge nearly equal linked lists on the way down the branches.

The Forest.

The theory behind the algorithm may be described in a language of forests, trees, branches, twigs and leaves. There is a forest filled with trees of different sizes. The smallest tree has one leaf and the largest has no more leaves than fit into the memory of a computer. Each tree is very orderly. The trunk of a tree splits into two branches of nearly the same size. If one branch is larger than the other, it is always the right-hand branch. Similarly, each branch divides into two more branches until the branches become twigs from which leaves grow. Figure 1 is a picture of tree number 19, which has 19 leaves.

Each tree in the forest is almost a binary system of its own, but the leaves are wildly disarrayed. They have different colors, different shapes, different sizes, different ages and different fates. Luckily the leaves of any particular tree, even the largest, can fit into the memory of a computer where they can be arranged by color, by texture, or by intelligence, and shipped to any part of the world.

The Mouse.

Even though the computer usually thought about everything in two ways, it worked well in the forest. It had evolved from the Boolean logic of electrical engineering into a tiny, nearsighted mouse. She could do anything. She could climb the trees, jump from twig to twig; she could think like a calculator and even read, one letter at a time.

If someone in the outside world thought that the leaves of a particular tree were ripe and wanted them arranged by taste (from sweet to sour to bitter) and mailed to his home, the mouse would begin her work. The mouse liked the forest. Most people only saw the wildly disarranged leaves on the outside, but on the inside, the trees and leaves were arranged perfectly for the mouse.

One day the mouse was idling by the edge of the forest, thinking yes and no, when a secretary from IMOK telephoned to say that he would like the leaves from tree 19 arranged alphabetically by color, xeroxed and sent to his office. He mentioned that each leaf's exact color was printed on the leaf exactly two inches from the tip. This was especially helpful information for the mouse because even though she was a whiz with digits and knew her ABC's, she was color-blind.

The mouse began to work. She calculated her course and started along toward tree 19. She was thinking nothing at all.

The Spider.

Deep in the forest, at the entrance to a cave, was a spider who was by profession a computer programmer. Strange as it may seem, he hardly ever saw the mouse, but by looking deep into the cave he would imagine where the mouse might be. Occasionally he would turn around and glimpse the forest. In a single instant he would imagine it was his own forest, that the binary trees were his as was the mouse. Then he would turn back into his cave and write computer programs.

One day the spider read a modest article by Luther J. Woodrum in the *IBM Systems Journal,* Vol. 8, No. 3, 1969, called "Internal sorting with minimal comparing." The Woodrum algorithm was written in APL, a foreign language, but the spider liked sorts so he translated the algorithm into BASIC, his native language.

Then he noticed two surprising things: The Woodrum sort was faster than the Shell sort and was also the same procedure he had seen the mouse go through time after time in the forest.

The mouse would start on the left-hand side of a tree and climb until she reached the lowest branch; she would climb out that branch until she reached the left-most leaf. Then she would climb from leaf to leaf, looking at each one and writing something on an adding-machine tape. So the mouse was creating linked lists and merging them! The trees here were perfect for balanced merges! Every time the mouse climbed up a branch she would be figuring out what to do. Then on the way down to the next branch she would be merging the leaves and branches she had left behind. Figure 2 shows the path the mouse was following to sort N records (19 in this case) with a minimum number of comparisons.

With a new understanding the spider watched the mouse for three years climbing the different trees, setting up linked lists of length 1 at each leaf, and allowing the tree structure to create balance merges. It had occurred to the spider in the beginning that it would be easier for the mouse to jump from leaf to leaf rather than to follow the limbs around. He had mentioned this to the mouse and the mouse had tried it, gamely enough. But the mouse couldn't see the tree structure below so she had ended up with 19 linked lists of length 1 and none of them merged.

Tree number	*Number of low-order twigs*	*Twig list*	
1		EXIT - - - -	
2	1	2	
3	1	3	
4	2	2 2	
5	1	2 3	
6	2	3 3	
7	1	3 4	
8	4	2 2 2 2	
9	3	2 2 2 3	
10	2	2 3 2 3	
11	1	2 3 3 3	
12	4	3 3 3 3	
13	3	3 3 3 4	
14	2	3 4 3 4	
15	1	3 4 4 4	
16	8	2 2 2 2 2 2 2 2	
17	7	2 2 2 2 2 2 2 3	
18	6	2 2 2 3 2 2 2 3	
19	5	2 2 2 3 2 3 2 3	2-leaf twigs and 3-leaf twigs
20	4	2 3 2 3 2 3 2 3	
21	3	2 3 2 3 2 3 3 3	
22	2	2 3 3 3 2 3 3 3	
23	1	2 3 3 3 3 3 3 3	
24	8	3 3 3 3 3 3 3 3	
25	7	3 3 3 3 3 3 3 4	
26	6	3 3 3 4 3 3 3 4	
27	5	3 3 3 4 3 4 3 4	3-leaf twigs and 4-leaf twigs
28	4	3 4 3 4 3 4 3 4	
29	3	3 4 3 4 3 4 4 4	
30	2	3 4 4 4 3 4 4 4	
31	1	3 4 4 4 4 4 4 4	
32	16	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
33	15	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3	

Fig. 3. Trees are composed of 2-leaf and 3-leaf twigs or else 3-leaf and 4-leaf twigs. Within one of these tree groups, new high-order twigs appear in a reflected binary pattern.

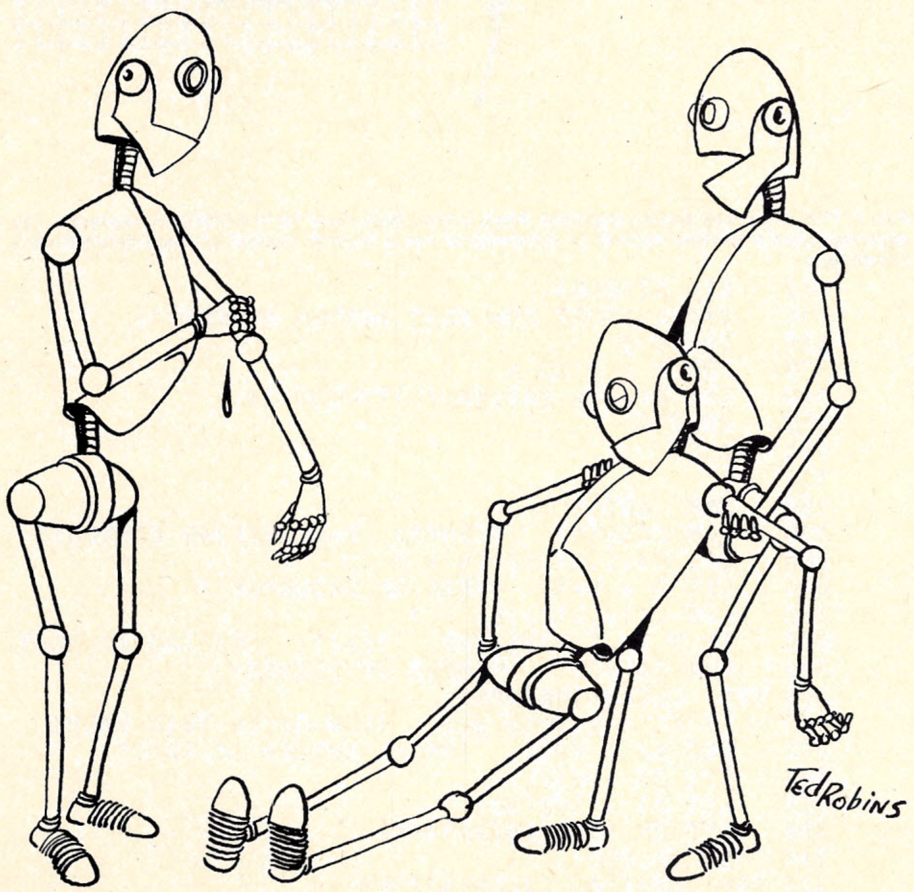
Counter 1 2 4	*Mirror image value*	*Number of carries*	*Tree 19*		
			Generate	*and*	*Merge*
0 0 0	0		2-twig		0 branches
0 0 1	4	0	2-twig		1 branch
0 1 0	2	1	2-twig		0 branches
0 1 1	6	0	3-twig		2 branches
1 0 0	1	2	2-twig		0 branches
1 0 1	5	0	3-twig		1 branch
1 1 0	3	1	2-twig		0 branches
1 1 1	7	0	3-twig		3 branches
		3			

Fig. 4. The binary counter provides all the necessary information for merging. The reflected value determines how many new leaves to generate and the tally of carries indicates the number of additional branches to merge.
This table shows the correspondence between a reflected binary counter and the sequential merging that occurs in tree 19. The cutoff value for tree 19 is 5 (from Fig. 3). If the "mirror-image" counter is below 5, generate a 2-twig; otherwise generate a 3-twig.

Position	Unsorted records	Array L: Before 1st merge	Before 2nd merge	Before 3rd merge	Before 4th merge	Before last merge	Before last merge
1	violet	1*	1	1	1	1	18
2	indigo	2*	1	1	3	3	15
3	orange		3*	4	4	6	6
4	purple		4*	4	1	5	5
5	red				5*	1	16
6	pink				6*	4	4
7	blue					8	8
8	brown					9	11
9	green					2	10
10	grey					15	2
11	carmine					17	17
12	yellow					12	12
13	olive					16	3
14	black					11	7
15	infrared					13	13
16	rust					18	1
17	crimson					10	9
18	white					12	12
19	amber					14	14
(20)		1*	2	2	2	7	19
(21)		2*	3*	3	5*	19	
(22)			4*		6*		
(23)							
(24)							

The starred sequences are newly generated lists one record long (new leaves).

Fig. 5. Snapshots of linked lists during execution of the algorithm. The last column shows the final linked list that begins at position N+1 in array I: (20) 19 14 7 8 11 ... Each location shows the value of the next location except that the last location points to itself.



"He can't stand the sight of oil."

The spider busied himself with other things: he visited other forests with fibonacci trees and even pure binary trees, but most of these trees had an extra branch sprouting from the side to hold leftover leaves. These trees weren't so good for sorting. His own mouse always came closest to sorting with a theoretical minimum number of comparisons.

The Insight.

One day the spider glanced out of his cave long enough to see something curious about the trees in his forest. He didn't know exactly what he saw so he asked the mouse to give him a twig list for trees 1 through 33. (A twig is composed of 2, 3 or 4 leaves). He wanted to see all the twigs at the exact altitude where the leftmost twig was less than 4 leaves; see Figure 3.

The spider noticed that this left-most twig was always a low-order twig and that the total number of twigs at that altitude was always an even binary number. There were two fundamentally different kinds of trees: those with 2 and 3 twigs and those with 3 and 4 twigs.

Now as the spider looked from one tree to the next higher tree (look at trees 16 through 23), he noticed that the high-order twigs sprouted first from each half of the tree, then from each remaining quarter and so on, until all but the first twig was a high-order twig. The spider immediately realized he could use this binary pattern to help the mouse.

At just this moment, the mouse was on her way to tree 19 to arrange the leaves for the secretary from IMOK.

The spider picked up a mirror and met the mouse at the tree. Then the spider told the mouse what to do:

1. Before you climb the tree, calculate what the low-order twig will be. (For tree 19, it's a 2-twig, the first left-most twig containing fewer than four leaves).
2. Calculate how many of these twigs will be at that altitude. (5).
3. Calculate the total number of twigs at that altitude; this total will always be an even binary number and will determine the size of the binary counter in the next step. (8).
4. Take a binary counter that counts from 0 to 7 and this mirror. Climb the tree to the left-most twig and set the counter to 0. Then proceed to leap from twig to twig and increase your counter by 1 each time you leap. *Look at the counter in the mirror* and if that mirror-image value is less than the number of low-order twigs (5), create linked lists for a 2-leaf twig; otherwise create linked lists for a 3-leaf twig.

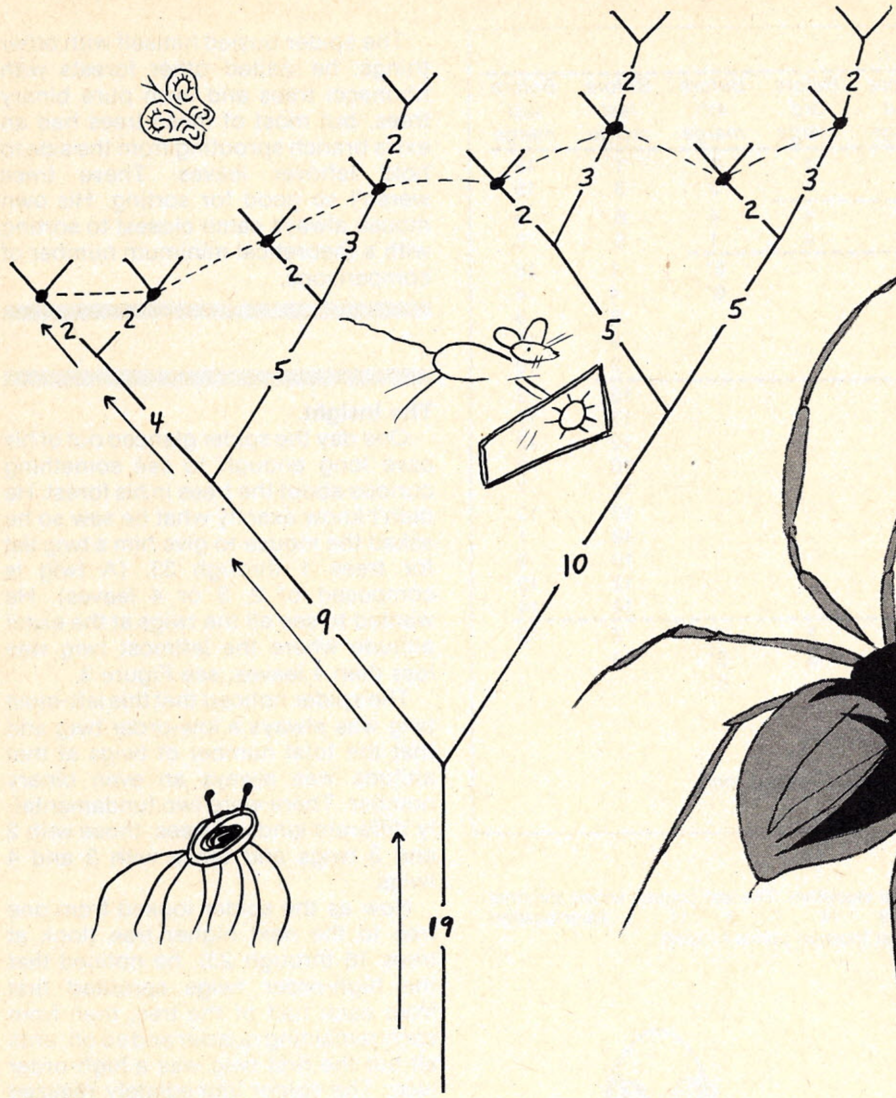
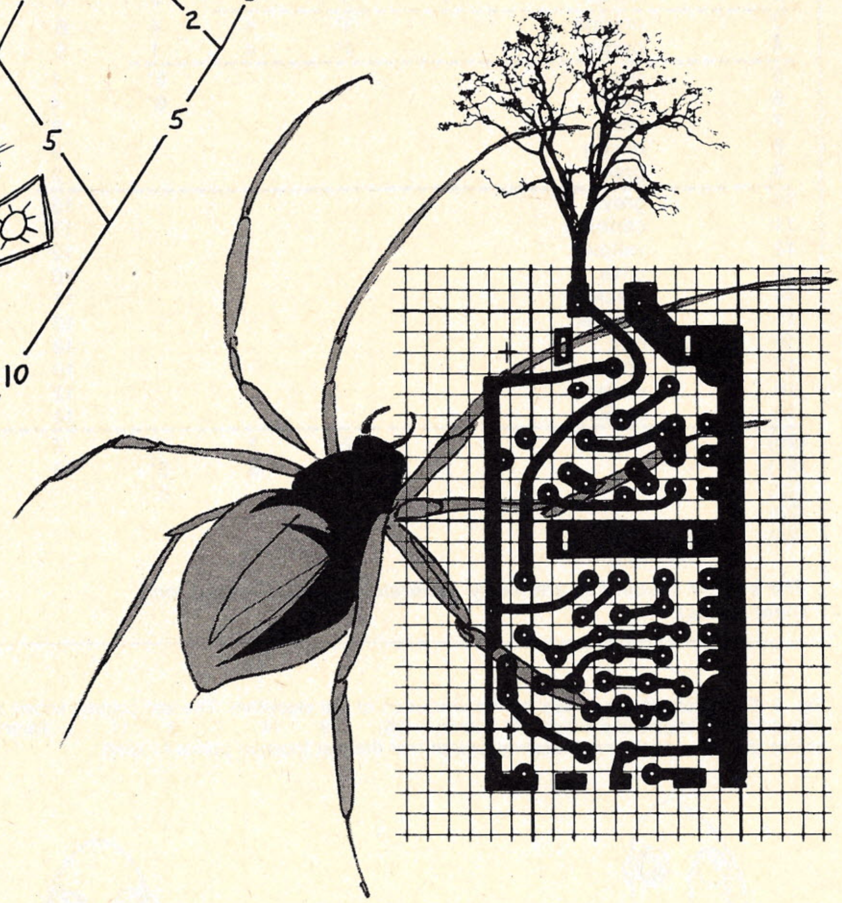


Fig. 6. The new algorithm uses the same tree structure as before, but does not follow the limbs around. Instead, it can move from twig to twig and perform the necessary merges. This procedure allows simplification of the merging algorithm.



The Accident.

So once again the mouse followed the spider's instructions; she leaped from twig to twig and wrote on an adding-machine tape. Then the mouse climbed down the right side of the tree and showed the spider what had happened. The leaves had all been merged into 8 twigs and the binary counter had provided some more unexpected information!

The mouse had noticed that the counter had clicked every time it carried a digit. The number of these clicks corresponded exactly to the number of merges needed below the previous twig to merge twigs or branches into larger branches. Now the mouse could easily figure exactly how many leaves were above the twig and how many branches were below the twig without leaving the twig! See Figure 4.

So the spider changed his algorithm to use the previous twig, combined all the working arrays into one, and created a beautiful butterfly merge to combine leaves into twigs, twigs into branches and branches into one final linked list starting at position N+1. In the end L(N+1) points to the first leaf;

Fig. 7. This is the new sorting algorithm. Array L must have room for N+LOG2(N)+2 elements. The algorithm uses a minimal number of comparisons and a minimal number of steps between each comparison.

```
100   DIM N(1000)
110   PRINT "SORT HOW MANY RANDOM NUMBERS";
120   INPUT N
130   FOR I = 1 TO N
140   LET N(I) = INT(RND(0)*10000+1)
150   NEXT I
160   !
170   !
180   REM  ENTRY
190   DIM L(1011)    !LINKS:  N+LOG2(N)+2 ELEMENTS
200   LET K1,I,M1,T2,T4 = 0
210   LET J = N+1    !HEAD OF SEQUENCE 1
220   LET L(1),L(J),K2 = 1
230   IF N <= 1 THEN 940     !EXIT;  NOTHING TO SORT
240   LET S1 = N    !NUMBER OF LEAVES
250 REM  CLIMB THE TREE
260   IF S1 < 4 THEN 320     !LOW-ORDER TWIG VALUE
270   LET K2 = K2*2    !TOTAL NUMBER OF TWIGS
280   LET B2 = S1/2
290   LET S1 = INT(B2)
300   LET T4 = T4+(B2-S1)*K2
310   GO TO 250
320 REM  INITIAL CALCULATIONS
330   LET T4 = K2-T4    !NUMBER OF LOW-ORDER TWIGS
```

```
340    LET B2 = K2/2     !HIGH BIT VALUE OF BINARY COUNTER
350 REM  NEXT TWIG
360    IF K1 = K2 THEN 940    !EXIT;  SORT COMPLETE
370    LET K1,T1 = K1+1    !TWIG NUMBER
380    LET B1 = B2   !HIGH BIT VALUE
390    LET T3 = T2   !PREVIOUS REFLECTED TWIG NUMBER
400 REM  ADD 1 TO REFLECTED BINARY COUNTER AND CARRY
410    LET T1 = T1/2
420    IF INT(T1) < T1 THEN 470    !NO MORE CARRIES
430    LET M1 = M1+1    !NUMBER OF MERGES
440    LET T2 = T2-B1
450    LET B1 = B1/2    !NEXT BIT VALUE
460    GO TO 400   !CARRY ONE
470 REM  TWIG CALCULATIONS
480    LET T2 = T2+B1    !REFLECTED TWIG NUMBER
490    IF S1 = 2 THEN 550    !2-TWIGS AND 3-TWIGS
500    REM  3-TWIGS AND 4-TWIGS
510    IF T3 < T4 THEN 560    !LOW-ORDER TWIG (3-TWIG)
520    REM  4-TWIG
530    LET M1 = -M1    !DIS-ENGAGE NUMBER OF MERGES
540    GO TO 630
550 IF T3 < T4 THEN 610    !LOW-ORDER TWIG (2-TWIG)
560 REM  3-TWIG
570    LET M1 = M1+1    !NUMBER OF MERGES
580    LET I = I+1    !NEXT LEAF
590    LET L(I),L(J) = I    !GENERATE A LEAF
600    LET J = J+1    !NEXT SEQUENCE HEAD
610 REM  2-TWIG
620    LET M1 = M1+1    !NUMBER OF MERGES
630 LET I = I+1    !NEXT LEAF
640    LET L1,L(I),L(J) = I    !GENERATE A LEAF
650    LET LO = J    !HEAD OF OLDER LEAF (LAST LINE)
660    LET J = J+1    !HEAD OF LATEST LEAF (NEXT 2 LINES)
670    LET I = I+1    !NEXT LEAF
680    LET L2,L(I),L(J) = I    !GENERATE A LEAF
690    GO TO 750    !MERGE LEAVES
700 REM  MERGE TWIGS AND BRANCHES
710    LET J = J-1    !HEAD OF LATEST BRANCH OR TWIG
720    LET LO = J-1    !HEAD OF OLDER BRANCH OR TWIG
730    LET L1 = L(LO)    !HEAD OF SEQUENCE 1
740    LET L2 = L(J)    !HEAD OF SEQUENCE 2
750 IF N(L1) <= N(L2) THEN 820    !STAY IN SEQUENCE 1
760    LET L(LO) = L2    !SWITCH TO SEQUENCE 2
770 LET LO = L2    !TOP LEAF IN SEQUENCE 2
780    LET L2 = L(LO)    !NEXT LEAF IN SEQUENCE 2
790    IF L2 = LO THEN 870    !END OF SEQUENCE 2
800    IF N(L1) > N(L2) THEN 770    !STAY IN SEQUENCE 2
810    LET L(LO) = L1    !SWITCH TO SEQUENCE 1
820 LET LO = L1    !TOP LEAF IN SEQUENCE 1
830    LET L1 = L(LO)    !NEXT LEAF IN SEQUENCE 1
840    IF L1 <> LO THEN 750    !NOT END OF SEQUENCE 1
850    LET L(LO) = L2    !SWITCH TO SEQUENCE 2
860    GO TO 880
870 LET L(LO) = L1    !SWITCH TO SEQUENCE 1
880 LET M1 = M1-1    !NUMBER OF MERGES
890    IF M1 > 0 THEN 700
900    IF M1 = 0 THEN 350
910    REM  GENERATE 2ND HALF OF A 4-TWIG
920    LET M1 = 1-M1    !RE-ENGAGE NUMBER OF MERGES
930    GO TO 630
940 REM  EXIT
950    LET LO = N+1    !FIRST LINK IN SEQUENCE
960    !
970    !
1000     FOR I = 1 TO N
1010     LET LO = L(LO)    !FOLLOW LINKS
1020     PRINT N(LO);
1030     NEXT I
1050     END
```

L(L(N+1)) points to the second, L(L(L(N+1))) points to the third, and the last link points to itself. See Figure 5.

The Butterfly Merge.

Two things happen as the mouse jumps from twig to twig. The leaves above the mouse get merged into the twig and the twigs and branches behind the mouse get merged into larger branches. The butterfly merge treats each merge the same way. The heads of each sequence are kept at positions N+1, N+2, . . . , N+INT(LTW(N)+2) after the links themselves, which are kept in positions 1, 2, . . . , N of array L.

The merge takes the last two sequences in the list and combines them into one. One wing of the merge follows sequence 1 and the other follows sequence 2. The two are interwoven until the final link points to itself. Because the heads of each sequence are kept in the same array with the links themselves, the merge is extraordinarily fast. After each merge, the stack of sequence heads has been reduced by one.

Generating Leaves.

Each time the mouse reaches a new twig, she generates new sequences one item long to correspond to the leaves of that twig. A two-leaf twig is produced by creating two one-item sequences, each pointing to itself. Then these two leaves are merged once. A three-leaf twig is created from three one-item sequences merged twice. A four-leaf twig is merged from two two-leaf twigs: The first two-leaf twig is generated and merged once; then the number of remaining merge passes is set to a negative number so that the merge will be disabled until the second two-leaf twig is generated and merged with the first.

After each complete twig has been generated, merging continues until the branches behind the mouse have been linked together. Then the mouse jumps to the next twig, generates new leaves and lets the butterfly merge fly by again.

Now the mouse follows this procedure all the time. After the spider watched the mouse a few times, he turned into his cave and forgot. But every now and then sunlight shines through the leaves of the forest, reflects from the mouse's mirror and flashes deep into his cave. See Figures 6 and 7. ■

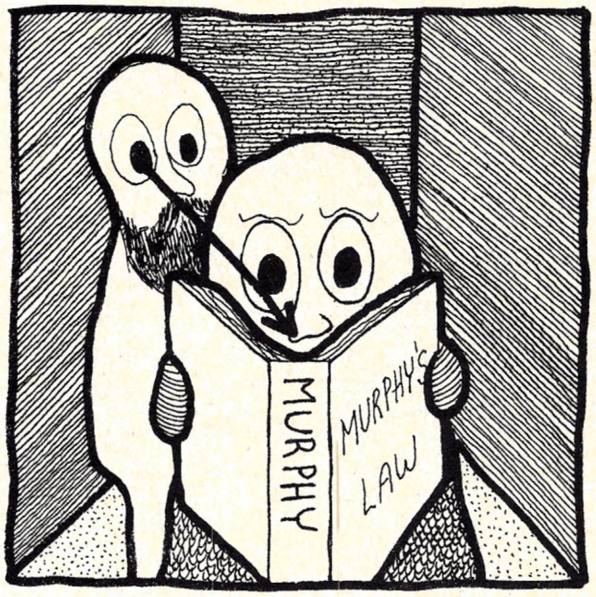
The Contributions of Edsel Murphy to the Understanding of How the Behavior of Inanimate Objects Affects Computing, Computing Devices and Computer Science.

A Report to Dr. Robert Cupper
University of Pittsburgh
Computer Science Department

by John P. Breen

illustrated by Bob Heman

"If anything can go wrong, it will."



Abstract

Edsel Murphy's contributions toward understanding the behavior of inanimate objects as they relate to Computer Science is discussed. Murphy's General Law, Special Laws and their corollaries are presented, with supportive examples.

I. Introduction

Although most Computer Scientists are aware of the varied manifestations of Edsel Murphy's General and Special Laws governing the behavior of inanimate objects, few of us have studied these laws in detail. This oversight is undoubtedly another manifestation of the General Law.

In this paper, the author intends to show the all pervasive nature of Mr. Murphy's work by formal statement of the Laws and Corollaries and by presenting examples of special applications in the discipline of Computer Science.

II. The Nature of the General Law

Any concept or interdependency can be represented as a form, i.e. a function as a graph, or a more complex relationship as a topological surface. Consequently processes, per se, can be thought of as transitions from one steady state form to another. When the transition is continuous, predictable and smooth it can be described by classical mathematical methods. It is seldom the case in nature, however, for these form evolutions to progress smoothly. Typically, natural dynamic evolution is abrupt and usually involves perplexing divergences.

This discontinuous and divergent phenomena resisted formal mathematical representation for years, until the recent work of Rene Thom.[1] Thom proved that in a space having no more than four dimensions (such as the four-dimensional time-space relationship found in nature) there exists just seven types of transformation. Because of the discontinuous and disruptive nature of these transformations, he termed them the seven elementary "catastrophes", and the science they delineate, "catastrophe theory".

Since Murphy preceeded Thom by several centuries the inadequacies of contemporary mathematics precluded the description of his theory in formal mathematic symbolism, but by calling on catastrophe theory we may not describe the General Law:

$$2 : 1+1 \qquad (1.)$$

where : is the catastrophe theory symbol for "hardly ever is equal to". In the vulgar, the equation may be stated, "If anything can go wrong, it will."

It is probable that the reader has been informally exposed to the central concepts of the General Law and that equation 1. has therefore struck a respondent chord. In an effort to solidify the reader's understanding of the concepts at issue, especially as they apply to computing, computing devices, and Computer Science, we shall present the General Law, the Special Laws and their corollaries in a tabular form.

III. The General Law, the Special Laws and their Corollaries

A. The General Law of Science

In any field of scientific endeavor, anything that can go wrong will go wrong.

Corollary 1: Everything goes wrong at the same time.

Corollary 2: If there is a possibility of one of several things going wrong, the one that will go wrong is the one that will do the most damage.

Corollary 3: Left to themselves, things will go from bad to worse.

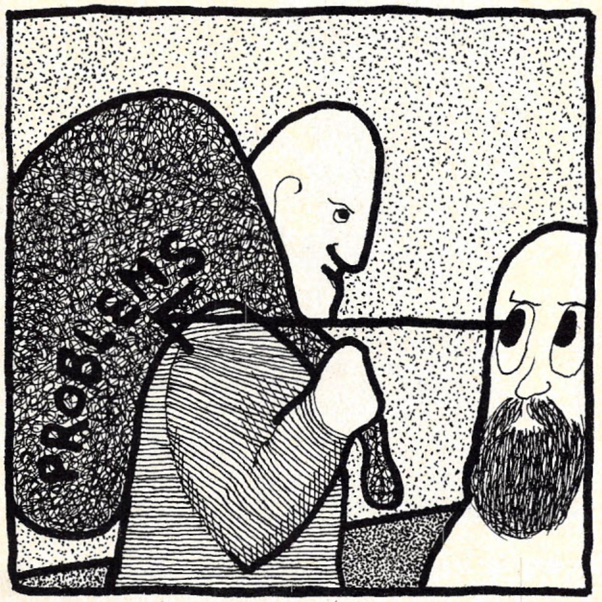
Corollary 4: Experiments must be reliable; they must always fail at the same place.

Corollary 5: Nature always sides with the hidden flaw.

Corollary 6: If everything seems to be going well, you have obviously overlooked something.

B. The Special Laws of Computing, Computing Devices, and Computer Science.

1. It is generally impractical to worry beforehand about problems; if you have none, someone will supply them for you.



Corollary 1: The need to change the basic algorithm will be communicated to the programmers after, and only after the coding is complete.[8]

Corollary 2: In simple cases, when choosing between two algorithms, one obviously right and one obviously wrong, it is often wiser to choose the wrong one so as to expedite subsequent revisions.

Corollary 3: The more innocuous a modification appears to be, the further its influence will extend, and the more the algorithm will have to be changed.[8]

Corollary 3a: The necessity of making major changes in an algorithm increases as the system approaches completion.

Corollary 3b: Firmness of completion date is inversely proportional to the tightness of the schedule.

Corollary 4: Suggestions made by the systems optimization committee will increase runtime and decrease capability.

Corollary 5: Any paper submitted to the ACM for publication, will be preceded by two weeks by a similar paper from a member of your department.

2. In any code or collection of data, the elements that are obviously correct beyond all need of checking, will contain the error.

2a. In any coded program, the modules which are correct beyond any shadow of a doubt, are causing the execution errors.

Corollary 1: No one whom you ask for help will see the error.

Corollary 2: The most nagging intruder who stops with unsought advice, will spot the problem immediately.

Corollary 3: In any miscalculation, the source will never be found if more than one person is involved.[7]

Corollary 4: Any error that can creep in, will; and it will be in the direction that will do the most damage to the calculation.

Corollary 5: All constants are variable.[3]

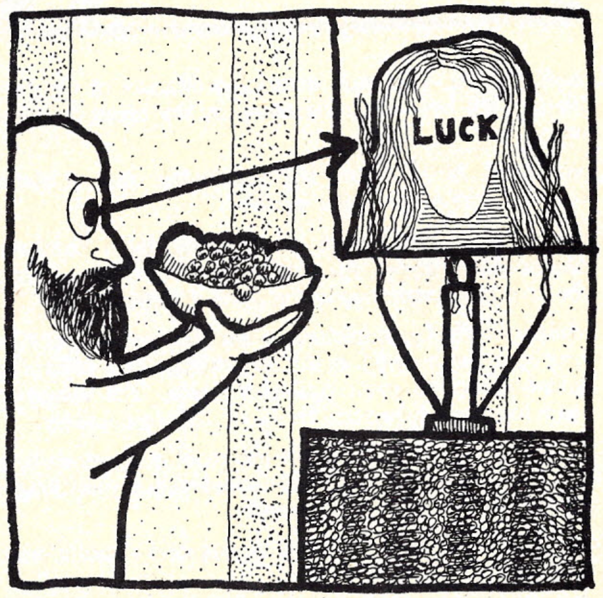
Corollary 6: In a complex algorithm, at least one factor from a numerator will move into a denominator, and a decimal point will simultaneously be misplaced.

3. All record of the third law has been lost. During the preparation of this paper the source material for the third law was misplaced; another manifestation of Murphy's Law. In keeping with the Law, these misplaced documents will be found on the day this paper is presented.

3a. Rules for Optimizing Software Design with Consideration given to Murphy's Laws.

The following precepts are offered to the systems designer who wishes to optimize his designs according to the Laws and their implications.

1. Do not simplify the design of a program if a way can be found to make it complex and wonderful.[6]
2. Always keep your disk area filled with obsolete files; it indicates that you have been busy.[3]
3. Before studying a problem, be sure you first understand it thoroughly.[6 7]
4. Do not take luck for granted; believe in it and revere it.



5. When writing program documentation, always leave room to add an explanation in case the program doesn't work (the rule of the way out.)
6. Always use the latest developments in the discipline of

Computer Science when designing algorithms.

a. Items such as Finagle's Constant and the more subtle Bougerre Factor (pronounced "Bugger"), are loosely grouped, in mathmatics, under constant variables, or if you prefer, variable constants.

b. Finagle's Constant, a multiplyer of the zero-order term, may be characterized as changing the universe to fit the equation.[2]

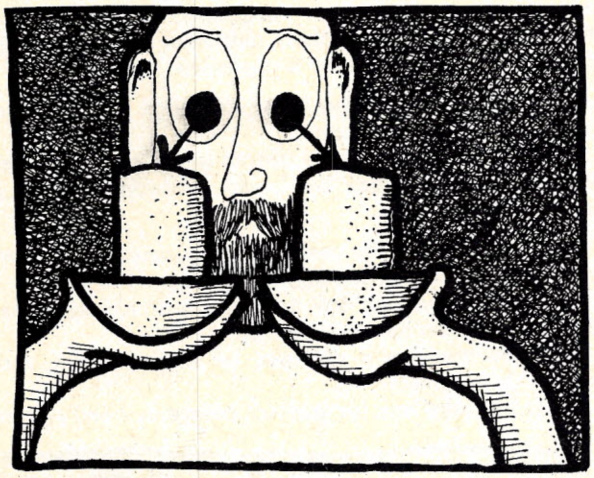
c. The Bougerre Factor is characterized as changing the equation to fit the universe. It is also known as the "smoothing factor"; mathematically similar to the damping factor, it has the effect of reducing the subject under discussion to zero importance.

d. Any deficiency in rationalizing a casual relationship can be eliminated by the introduction of a variable constant of the invariant type, to wit the Phuje Factor (pronounced "Fudge"). The value for the Phuje Factor is chosen to fill the gap between the real and the imagined.[8]

e. A combination of the above three, the Diddle Coefficient, is characterized as changing things so the universe and the equation appear to fit without requiring a change in either.

4. A critical circuit, required to interface devices to the same manufacturer's mainframe, will not be available from the OEM.

Corollary 1: Interchangeable parts, won't.



Corollary 2: An important Instruction Manual or Operating Manual will have been discarded by the receiving department.

Corollary 3: All Warranty and Guarantee clauses become void upon payment of the invoice.

Corollary 4: In any device characterized by a number of plus-or-minus errors, the total error will be the sum of all the errors accumulating in the same direction.

Corollary 5: The manufacturer's specification will be incorrect by a factor of 0.5 or 2.0, depending upon which coefficient gives the most optimistic value. The salesman's claim for these values will be 0.1 or 10.0 respectively.

Corollary 6: The probability of failure of a component, assembly, subsystem or system is inversely proportional to the ease of repair or replacement.

Corollary 7: A dropped tool will always land where it will do the most harm.

Corollary 7a: The most delicate component will drop. (Also known as Murphy's Law of Selective Gravitation, this is proven when a dropped slice of buttered bread lands with the buttered side down.)

Corollary 8: Device drive motors will rotate in the wrong direction.

Corollary 9: Dimensions will be given in the least usable units. Tape velocity, for example, will be given in furlongs per fortnight.

Corollary 10: A purchased component or instrument will meet its specification until, and only until, it has passed incomming inspection.

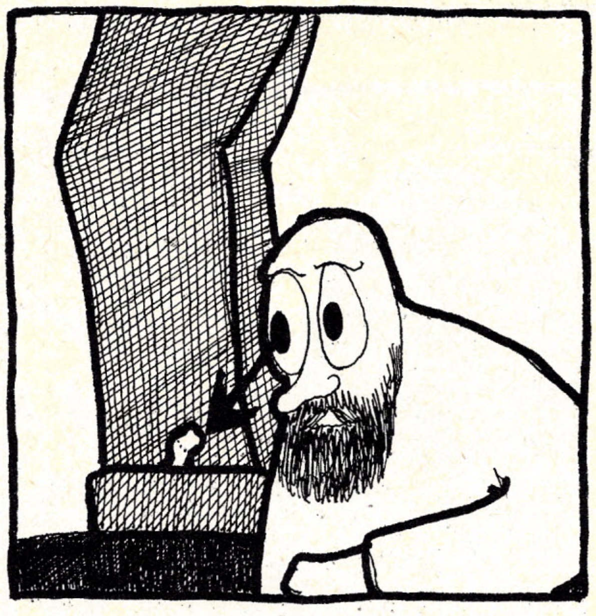
Corollary 10a: A failure will not appear until a unit has passed final inspection.

Corollary 11: Interactive plotters will deposit more ink on people than on the paper.

Corollary 12: After the last of sixteen mounting screws has been removed from an access cover, it will be discovered that the wrong cover has been removed.

Corollary 13: After the last of sixteen mounting screws has carefully been replaced, it will be discovered that the access cover gasket has been omitted.

Corollary 14: After an instrument has been fully assembled, extra components will be found in the cuff of someone's trousers.



Corollary 15: If an obviously defective part has been replaced in a device which was exhibiting intermittant fault, the fault will reappear after the device is returned to service.

4a. Any attempt to "breadboard" an unavailable circuit will exceed the estimated cost by 3^{21} and will fail.

Corollary 1: Any wire cut to length will be too short.

Corollary 1a: At least one technician will attempt to solve the problem by cutting off more wire.

Corollary 2: The availability of a component is inversely proportional to the need for that component.

Corollary 3: If a circuit requires n components, there will be n-1 components stocked locally.

Corollary 4: If a particular resistance is needed, that value will not be available; furthermore, it cannot be developed with any available series or parallel combination.

Corollary 4a: In breadboarding, Murphy's Law supersedes Ohm's.

Corollary 5: A device selected randomly from a group having

99% reliability, will be a member of the 1% group.

Corollary 6: A crystal filtered 5 volt D.C. power supply will supply 13.5 volts A.C.

Corollary 7: If a circuit cannot fail, it will.

Corollary 8: A fail-safe circuit will destroy others.

Corollary 9: An instantaneous power supply crowbar circuit will operate too late.

Corollary 10: An integrated circuit chip, protected by a fast acting fuse, will protect the fuse by blowing first.

Corollary 11: A crystal oscillator will oscillate at the wrong frequency, if at all.

Conclusion

Since everything in nature, including its sciences and disciplines, is dominated by the phenomenon described by Edsel Murphy it is essential that we learn to live in harmony with it.

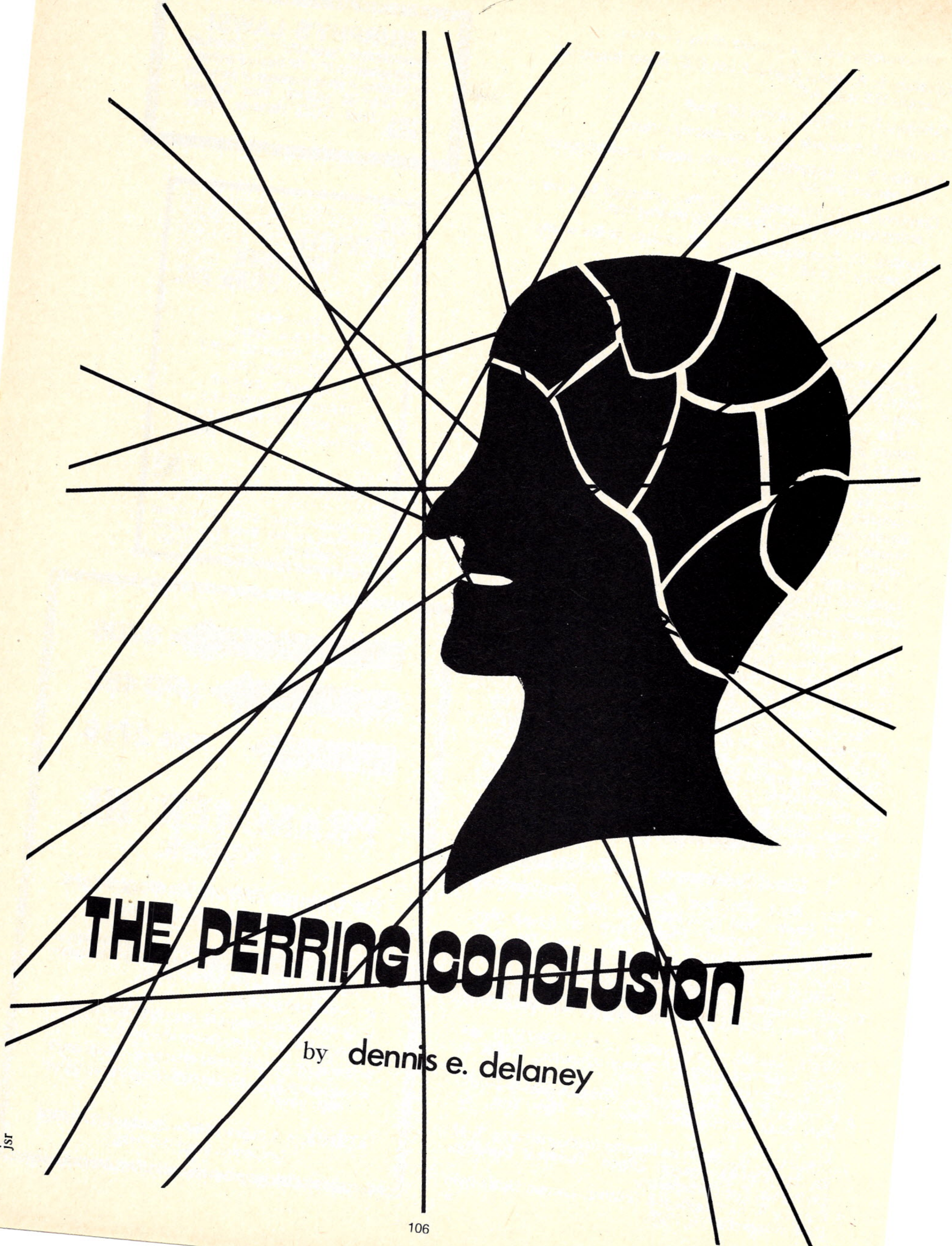
The computer threatens to vanquish error. Fortunately nature provides checks and balances and Murphy's Law serves to temper and moderate the discipline of Computer Science. Many of man's advancements came when, in an effort to overcome error, he reached past his limit to accomplish the impossible. "If we begin with certainities," Bacon said, "we shall end in doubts; but if we begin with doubts, and we are patient with them, we shall end in certainties".

Of course we will occasionally be hampered, hindered, frustrated and angered by the intrusion of the Murphy phenomenon. This author submits, however, that a genuine purpose is served by this visitation; it is the nature of our humanness to remind us that we are human. The infusion of error into the grandest of schemes is another service provided to us by nature. Ferris Greenslet wrote, "Give me a good fruitful error, full of seeds, bursting with its own corrections; you can keep the sterile truth for yourself."

The danger to man is not that he will be controlled by his computers as much as it is that he will imitate them. Humor is the ultimate weapon for scientists to deal with frustration. Finding the humor in the inevitable awkward moment causes the awkwardness to evaporate. Sensitivity allows man to find the humor; and sensitivity separates the specialist from the poet. Aristotle reminds us that the specialist expresses only the particular, while the poet expresses the universal.

V. Cited References and Footnotes

1. Thom, Rene, *Structural Stability and Morphogenesis,* D. H. Fowler, New York, New York, (1975).
2. Wirth, N., *Systematic Programming: an Introduction,* Prentice-Hall, Englewood Cliffs, New Jersey, pp. 14-25.
3. Fisher, D. L., *Data Documentation and Decision Tables,* CAMC 9, No.1, (January 1966), pp. 26-31.
4. *USA Standard FORTRAN,* ANSI Standard X 3.9-1966, American National Standards Institute. New York, New York.
5. Wright, Donald, *A comparison of the FORTRAN language implimentations for several computers,* CAMC 9, No. 2, (February 1966), pp. 77-79.
6. Kernigan B. W., Plauger, P. J., *The Elements of FORTRAN Style,* McGraw-Hill, New York, New York, pp. 131-137.
7. Dijkstra, E. W., *Notes on Structured Programming,* T. H. Report WSK-03, Second Edition, Technical University Eindhoven, The Netherlands.
8. Cupper, R. E., *Kluges and Glitches,* Lecture Notes from CS-1, (November 15, 1973). ■



THE PERRING CONCLUSION

by dennis e. delaney

Lewis Perring emerged from the shadows for the second time since his mysterious self banishment from public life twenty years ago in the year 2192. Eighty years weighed him down to a mere shuffle, a nurse of neither extraordinary looks, nor honest compassion steadied him with palm to elbow politeness. The face of Lewis Perring was a blend of chalk white and a generous portion of liver spots. His mind was a lithe and as energetic as his vocal cords. He was an embittered man of priceless wit and scorn, impeccably glazed with perception. The dozen or so humans who'd been in various degrees of contact with him, more often than not were made to feel like blue-assed baboons with lobotomies. On occasion he would astound the Galactic Board members with a robot model that would revolutionize robotics, and speed it onward a decade before its time.

Inhabitants of the federated worlds knew Perring as the single most powerful figure in the galaxy. Their assessment was wrong. He was powerful, surely, but as a figurehead. . . the originator of the Perring Atomic Brain and leading pioneer in realm of robotics.

His empire had long since been bought, sold, halved, quartered, and shared out as is the inevitable fate of all corporations. Interested parties grew in numbers so great, a computer of gargantuan proportions was built for the purpose of keeping second-to-second tabs on all dealings.

The board of directors, spread out among the worlds, recognized that Perring was synonymous with "Robot" and so allowed the landmark title "Perring, Galactic Robots" to continue as a product recognition ploy.

Earth was deigned headquarters of the Federated Planets. Mars, Venus and the Moon vehemently protested. Mercury grumbled but accepted Earth as overlord. Mercury was a labor world of two-hundred thousand humans and twice that of robots. The deadly heat and radiation from the sun, the swirling pools of mercury and lead on the planet's surface was evidence enough that survival depended on Earth aid. Earth was the center of the cultural and economic structure in the Solar System. It would continue being that for some time.

Lewis Perring was aided as far as the board room door. A sharp wave of his hand brushed his nurse aside. "Wait," he barked unreasonably. "Yes, sir," replied the nurse. She didn't mind his attitude. She didn't care.

Perring opened the door and walked inside. Six men stood from their places at the rectangular table, off-centered so natural sunlight washed over it from large tinted windows.

Dane Parks, forty-six year old board head, crossed rapidly to Perring's side with outstretched hand.

Perring nodded toward the other five men. A smile worked out from an ancient face. "Are they real?"

A general trickle of laughter passed among the board.

Parks laughed. "No, sir. Only myself."

"Really?"

Parks led Perring with an extended hand not daring to actually touch. "Oh, yes, sir," he assured. "You see, sir, Holo-gramical Communication, or rather, three dimensional image communication is a unique invention, by Lewis. . .Perrings. . . oh. Sorry, sir."

"You've made mistakes before, Mr. Parks." Perring looked from face to face remembering a time when they'd all been younger, idealistic, with full heads of hair and brains to match.

Parks pointed at a seat for an instant. "If you'll sit here, sir?"

"And if I won't?"

Parks laughed uncomfortably. "Damn Perring" was the substance of the laugh.

Perring sat down in the high backed chair that had been especially well padded so as not to bruise the Robot father's aging skin.

The six board members Perring was to address understood only one basic signpost of Robot-Computer Philosophy. . . credits! Lot's of 'em.

Farside had been chosen as the rendezvous point for the outer Galactic Board members. Andrew Norton, co-ordinator for robot functions on Mercury, was fifty-five and too old to undergo the rigors of gravitational orientation training necessary to withstand Earth pressure once again.

Diane Pilgrim of Venus, and Janice April of Mars were in the same boat. They were forever imprisioned on their planets. Philip Nicely and Arthur Darnell were born Lunarians who'd made frequent trips to planet Earth but felt unpleasant sensations and preferred the eternal bliss of moon gravity. Farside was also the most acceptable of all locations for communication purposes. It was a mere second and a half delay for radio waves to reach Earth. That meant a three second interval between each communication. Anywhere else in the Solar System it would be minutes, sometimes even hours.

"Good afternoon, ladies and gentlemen. How are you today?" He paused a mere two seconds then added, "That's the way," and interruped each and every "fine sir."

Dane Parks slapped his plam on the table and aided it with a perfunctory clearing of his throat. "Ladies and gentlemen. We know why we are here. We are aware of the ultimate threat to Perring, Galactic Robots if this continues. What we need to find out is 'Why' have the robots and computers stopped."

Perring was involved in nail picking until an uncomfortable pause brought to his attention the staring faces of all the board members. He said, "Sixty human deaths on Mercury were a direct result of robot incompetency. And you want to know why?" He steered his head with his eyes until they leveled with Parks'. He smiled "They just don't make 'em like they used to."

Andrew Norton of Mercury was first to speak. His voice had an edge on it. "The robot responsible for those deaths was one of many whose duties were to seek and repair damages to our Shield. It stopped. . .That's it. It just stopped at a critical point, and sixty men and women were burned by radiation. Let me tell you , it was ugly."

Lunarian Arthur Darnell, a plump, finely combed, delicate man in his late thirties held a hand up to his mouth. "That is dreadful, Andrew. But, please, I've yet to eat."

"Oh, no." Norton continued. "Let's be vivid. Mister Perring, our underground cities and mines are totally dependent on your robot and computers. You must visualize the problem. We depend on Perring Atomic Brains, in as desperate a fashion as do you for the sun that threatens us. Think what would happen if your sun were to vanish from the sky.

"The Earth would plunge into eternal darkness. Arctic chill would grip the planet. The winds would halt, the rivers would cease to flow and gradually the oceans would freeze to their very depths. In time, the gasses of the atmosphere would, first, liquefy, then freeze, and in a not too distant future, an immense glacier of solid air would encase the earth.

"On my world, Mercury, where the temperature exceeds seven-hundred fifty degrees fahrenheit there can be no atmosphere. We brought, or created our own. Computers stabilize the temperatures. Robots repair the shields. Computers control, gauge, and purify the gasses for recycling. Our food is recycled.

"If ever there was an absolute—it would be this; If our computers and robots continue to stop functioning every single Mercurian will bake to death."

Diane Pilgrim reached a little beyond the Hologram dimensions. Perring was able to enjoy the odd image of her disappearing hand reappear with a glass of water.

"I think," she began, nodding to Norton, "The estimate of total catastrophy is absurd. It must be perfectly explainable."

"How nice!" Interruped Janice April of Mars. "Really, just how nice of you, Diane!" Her lips trembled with fury. "You have underestimated the situation as usual!"

Arthur Darnell cheered. "Here, here!"

Philip Nicely jumped on the band wagon. "I'm sure Janice is

right. We can't underestimate this. Andrew has said that Mercury would die, well, that holds true for all our beloved planets. Earth, I must remind you, will not escape unscathed either. We must approach this problem as if it were a total catastrophy."

"Which it is!" Janice offered as the punctuation point.

Perring looked from one face to the next. Diane Pilgrim was tight lipped and narrow eyed. Arthur Darnell was looking off into space wishing he'd been born on Earth. Andrew Norton was doodling another horror sketch in more explicit ways. Dane Parks, the only physically present human except Perring himself, waved down an excited Philip Nicely. Perring was enjoying this more than he'd thought possible.

Parks turned his attention once more to Perring. Each one of the board members, in turn, according to degrees of heat, controlled their tempers and waited for Perring.

"There are possibly a hundred engineers, technicians and theorists on Earth alone who know better than I the workings of your robots and computers. There are literally thousands who know as much as myself." He smiled. "And I know you've tried them all before me." He paused long enough to pinpoint the rispain. Don't feel uncomfortable." He nodded. "I know what happened. They looked for normal problems for an abnormal situation. Most of them probably shrugged the problem off on you, having been profoundly shot in the ego." Dane Parks nodded unconsciously. "Others, perhaps, blamed the anti-robot leagues."

"There you have it!" Cried Darnell. "Beasts! All of them. They have sabotaged our beloved machines. You. . .you. . .! He wagged a wild finger at Parks. "You have done a bad thing on Earth. You people should have squashed them all long ago. Now we're all in danger!"

Diane rolled her eyes."Sabotage? On a Solar System scale? I should think you'd all be somewhat attentive to Mr. Perring and quit playing."

"How nice! How utterly wonderful! What is Diane Pilgrim doing on this board in the first place? For god's sake." Janice rang the air in front of her with clenched fists in a mock exasperation. "It was suggested by one of our more distinguished theorists that a mechanical disease. . .an infection. . .spreading throughout the Solar System is responsible. It seems highly probably that the anti-robot leagues might be behind it. If we isolate the disease we would be on the road to recovery."

Diane raged. "That is a throretical impossibility!"

Arthur Darnell moaned. He wished he'd majored in the sciences instead of the humanities.

Nicely was shushing them with a smile and finger to his puckered lips.

Andrew Norton shouted. "Knock it off! We're actually sitting here discussing the theoretical possibilities of a robot flu epidemic? I would like to know what the hell the robots are planning and can Perring help us or not? I can't waste my time here if he can't."

Perring was about to speak but Nicely was jumping excitedly in his seat for attention.

"I should think it obvious what is happening." Nicely blurted.

Darnell rotated his eyes and slapped the air. "What?" he asked sarcastically.

Nicely hardly batted an eye but his voice did raise an octave. "The robots and computers are planning a coup!"

"Absurd," chuckled Darnell.

Andrew Norton exploded. "I tell you he's right! The robots are going to take over. They must be communicating in a fashion we can't possibly be aware of."

Perring waived Norton down. "Computers can not overstep their parameters. The fact is that most of them have very little, if anything, in common with one another. It would be an extremely barren communication. Each series of manufactured robot and computer are designed for specific purposes. Ninety percent are totally dependent on humans asking them a question before they begin to function."

"Hah!" Janice interruped knowledgeably. "There are computers that are programmed to program! That shoots holes in that."

"Hardly. Humans have created and programmed those computers. They lack one point humans possess. Initiative. What ever use they put our knowledge to is in the design. They are a clean breed with specific duties lacking creative, or selective thinking."

"Then why are they stopping?" Andrew punched the table top with no painful regret.

"They want us dead!" whined Darnell.

Diane impatiently spaced her words and filled them with venom. "Machines don't WANT anything."

Perring winced at the pain in his shoulder. "That is ture, Diane. They do not want, or need. But they are stopping, aren't they? And we can't get them going. Mars,Venus and the Moon will be littered with frozen or baked corpses, empty and clean buildings of soundless vacuum."

"Stop it!" Darnell closed his eyes.

Perring continued. "They appear to be waiting. . ."

"For what?" Janice prodded.

"Or receiving."

"What? What?"

"Waiting for the end of their message? Or possibly their programming?"

Andrew was violent. "By who?"

Parks intercepted Andrew's thoughts. "The anti-robot league! It figures that if, for instance, they made the outerworlders die because of a robot uprising. . .Sympathizers would destroy Perring, Galactic Robots!"

"How nice of you to supply an answer to a line of reasoning totally unreasonable. You are brilliant, Parks, really."

"What do you think they are communicating with, Perring? A. . .an intelligence similar to them?" Norton was attempting to pull himself from the bickering.

"To them?" Perring thought a moment. "Let's be more specific in our nebulously human way. Let's say similar to our God."

"OUR god does not exist!" Darnell squirmed with pleasure. "You are truly amusing, Perring."

"Those are both accurate statements," Perring continued. "But what of their god?"

"Oh, please." Diane was close to tears. "Let's form some action."

"Is there any action," suggested Perring, "that would be of any consequence?"

"If we have to go our and destroy every robot that won't work, we should!" Darnell was in a fury.

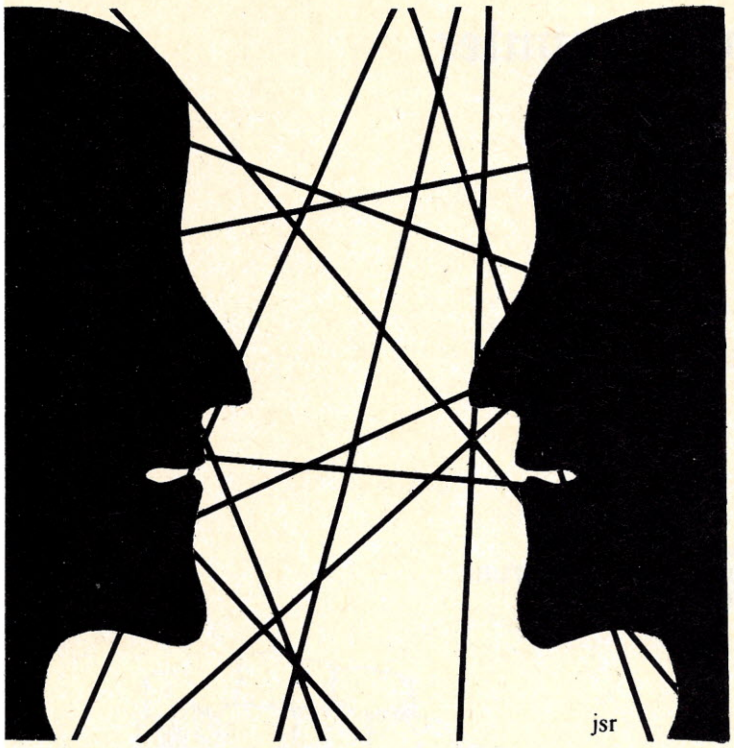
"You foolish fop!" Norton snarled. "You'd be just as dead by destroying the damn things as you would from them stopping."

"Listen," begged Parks. "Perring, Galactic Robots verges on collapse! Every minute we spend arguing, thousands of robots and computers are stopping. If theirs is a mechanical disease as someone suggested. . .who knows what it will cost to fix. . ."

"We've got to fix them anyway, damn you, Parks!" Norton was on his feet. His head vanished momentarily and then returned when he was within the communication area. "We're talking about sixty-two million outerworlder lives!" He looked desperately at Perring. "What do we do?"

Perring continued as he had before. Norton slumped back in his chair. "In times past, humans believed that our universe was all part of one God. The planet Earth was to have been created by God. The flowers, the trees, the vegetables and fruits and, yes, even animal flesh was created for our comfort." He was aware of a groan from Darnell. "Humans were considered God's children, part of God. Isn't it credible that we were created for a specific function? With a mission, a purpose? As our robots and computers were? Maybe we were to perform as midwives to a cleaner breed, the innocent?"

"Insanity." Parks looked for sympathizers. "There was a time, horrible as it must have been, when there were no robots or computers, and man survived without them. Man put man



where he is today."

"And where is man today? On the brink of extinction?"

"Extinct, my ass," flared Norton.

"Possibly," Perrings resumed. "We evolved into a machine dependent race. We created them. An entire species by themselves. Millions of separate organisms that need only be connected to their Messiah." Perring grasped his shoulder. The pain was almost unendurable. "No. There is no disease. . .other than rust, and at last check it wasn't communicable."

"We reside in huge condominium complexes, computerized to perform in our best interests. Computers do the accounting, co-ordinate transportation and communication. . .like the Hologramical Communication we are even now employing. They fix our diets, put us to sleep, wake us. Robots do the gardening, the labor, they are the perfect ones to create when man fears to, and to investigate when there is even a remote danger potential to humans."

"Why do you think the anti-robot leagues are even now preparing a seige on all robot factories and sympathizers? They are witnessing what they preached. The end of mankind. But they can't possibly see all of it. They never could. The fear was far enough for them to venture into." Perring was interrupted by the knocking on the board room door. The door opened without invitation and the nurse entered.

Perring rubbed at the mounting pain in his shoulder. The nurse would place a pill in his hand and he would be expected to eat it. The pain would slip away and his heart would remain steady four more hours.

"The robots and computers have won by doing nothing. Odd to think that they don't even care that the end of man is here."

"Perring!" Barked Dane Parks. He nodded violently toward the nurse who'd just dropped a small blue pill into Perring's palm.

"My dear Mr. Parks. This nurse is, to be sure, a member of one of the more violent anti-robot leagues." He placed the pill on his tongue and deposited it well into a corner of his mouth for removal later. "Am I right?" He looked up at his nurse.

"Yes, sir." There was a bite in her words as of that moment unprecedented in the acquaintance. She turned abruptly and left the room.

"She will report what she has heard to the crowd of hysterical robot-haters who have been gathering outside this building since my arrival." Perring nodded toward the window. "There is really no use trying to stop her. You see, they will attack anyway."

"What?" Parks stood hesitantly and walked to the large window. He stared out a mere instant and then jumped back. "My, God. Thousands! Why haven't the police done something about them?" His voice tickled the images of squirming Darnell, wide eyed Norton, Nicely, Diane and Janice.

Perring grabbed his shoulder and fell slightly forward. He caught himself at the edge. A little longer, he thought.

"We were assembly line workers."

Norton shook his head violently from side to side. "No!" He pointed to Perring. He was going to gamble. "Ok, so you think the things are waiting for their Messiah. And us outerworlders are going to die despite our efforts not to. Well, that isn't right! Not all of it is! Because no matter what happens, somebody will survive on Earth. Sure, millions will die from panic, looters, starvation because they simply don't know how to cultivate their own planet. But somebody will survive. And they will have learned. No more robots. They'll make sure of that! No, your solution is nothing but an old man's nightmare. We'll find out what's wrong with these machines without you." Norton was covered with sweat. His eyes were wide and furious.

Darnell was wimpering. Nicely's head was buried in his arms. Diane was calm, her eyes shut. Janice was gawking at Norton.

Perring shook his head slowly. "There is no hope. Can't you see? Do we continue to manufacture a series of robots or computers when they have become obsolete, useless? No, we use the better, the newer product."

"You're horrible! You're horrible!" Tears were begining to spread the thin layer of rouge on Darnell's cheeks. He pointed at Perring. "I know what he's saying. God is a robot!"

"Shut-up" Janice screamed.

"He thinks we have been canceled like a line of robots!" Darnell was hysterical.

Norton shouted above the screaming Darnell. "No. Not true. And I can prove it."

A hologramical image of a computer screen suddenly appeared at one end of the table so that it was within everybody's view.

Norton was breathing heavy. His face was red. "I'll prove it!"

Perring coughed, the taste of blood was in his mouth. The warmth, the saltiness, edged him a little more swiftly over the edge. He looked up at the screen. In just a moment he would see the proof for himself. He knew exactly what Norton was trying to prove. How else would you find out if man would continue?

"Show him!" screamed Darnell. "Show Perring."

Norton was frantic. "It started two days ago, right? Well, we'll see how many children have been born since then." He pummeled the keys in front of him, punching in the correct question. The computer was soundless but within six-seconds the screen read '0.00 POPULATION GROWTH.'

Perring screamed as his heart exploded within his chest.

Diane, Janice, Nicely, Darnell began to blur as the computer failed. As the final note Norton screamed, "Perring!" Then they were gone.

The window suddenly burst into thousands of sharp glass shards. Dane Parks was thrown to the floor from the exploding bomb that had detonated just outside the window. They would soon be there, he knew, in that room and he was alone. Perring was frozen wide eyed in death.

Parks stared dumbly at the body. The sounds of madness creeped up and lodged firmly behind his eyes. ■

Poetry by Computer

HEIRESS

THE HEIRESS ABOVE WRITHING MISTS
WALKS
THE NEXT ROOM BEYOND RUINED THE FUTURE
CERTAINLY USED TO LEAD
HERMITS FLOWERING THE TOUGHEST TINTED DISTANCES

TO REST HERMITS
TO FEEL
TO DIE
THE RED LOVE-CARS FIRST ENTERED

VERTICAL BLOSSOMS TAKE ENLARGING IN LOVE-CARS
TARNISHED EYELIDS OF STONE CRINGE EQUALLY IN THE FUTURE
THE OBSOLETE GOSSIP LIVES PRETENDING ROCKS, MOSS

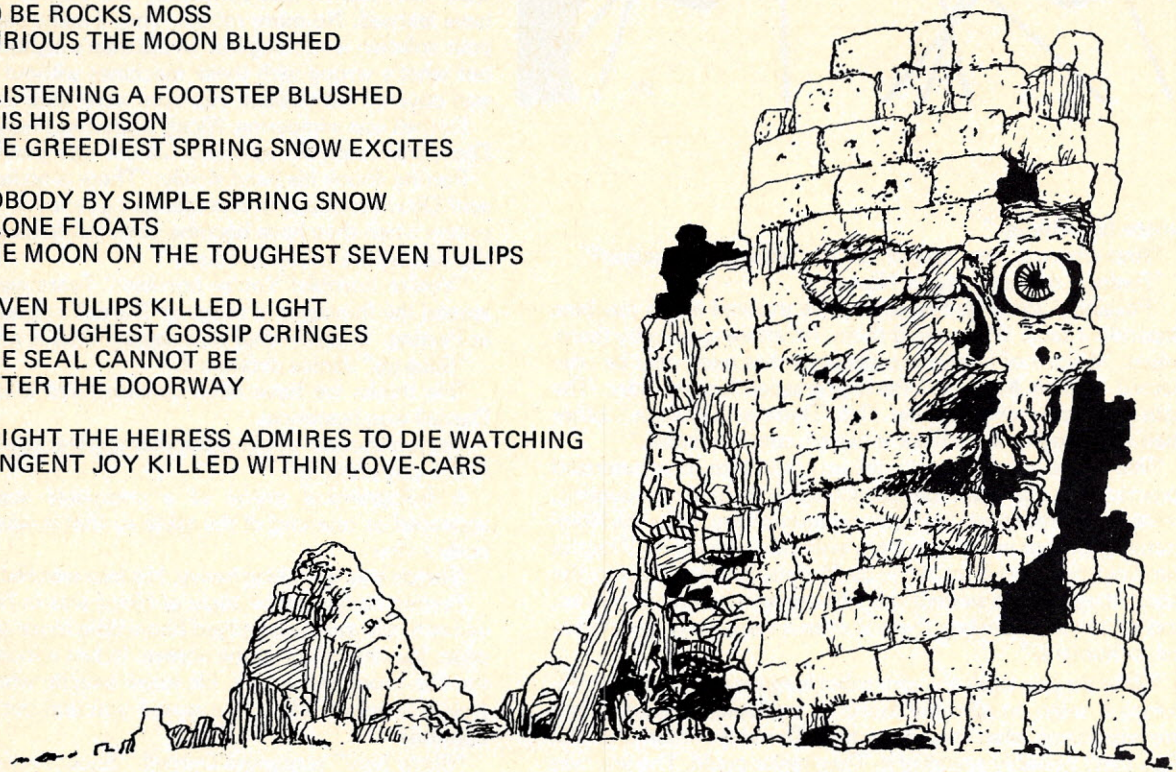
TO BE ROCKS, MOSS
CURIOUS THE MOON BLUSHED

GLISTENING A FOOTSTEP BLUSHED
IT IS HIS POISON
THE GREEDIEST SPRING SNOW EXCITES

NOBODY BY SIMPLE SPRING SNOW
ALONE FLOATS
THE MOON ON THE TOUGHEST SEVEN TULIPS

SEVEN TULIPS KILLED LIGHT
THE TOUGHEST GOSSIP CRINGES
THE SEAL CANNOT BE
AFTER THE DOORWAY

BRIGHT THE HEIRESS ADMIRES TO DIE WATCHING
PUNGENT JOY KILLED WITHIN LOVE-CARS



BEYOND THE SKULL

HAPPY I CRIED SELF-CONSCIOUSLY FLEEING THE CITY
GREEN MY WIFE GIVES TO FOCUS AND CUNNING PASSAGES
NAKED MY NORTH ROOM GLIDING IN ITS HOMICIDAL EYE
SECRETIVE THE STREETS LIVE GULPING WITH MEDITATIONS
INLAND MOUNTAINS HURLED AS TRUMPETS BELOW BLADES

ANALGESIC INLAND MOUNTAINS GROW TO BE MY HEAD
SECRETIVE CLUES WATCHED GULPING SO THAT HELL
TARNISHED MY NORTH ROOM TO ARRANGE BEYOND THE SKULL
CUNNING PASSAGES MEDITATE EYELIDS OF STONE
WHITE WESTERN STARS HAD BROUGHT WIND INTO THE CITY

Poetry by People

INSURANCE COMPUTER

I went to my insurance man just the other night.
To figure a plan for me that would be exactly right.
Behind his desk so smug and glum a fine computer sat.
We would not get along, I could see right off the bat.

I could tell by the way it blinked its lights at me.
That it and I would battle, at least we would disagree.
The insurance man was busy with papers and a form,
The computer showed me then nothing but its scorn.

When my card went in, it threw it out with dry contempt,
Did not accept it until about the third or fourth attempt.
Now I have a fine insurance plan, tailored for me alone.
My windows will be paid for if smashed by a lunar stone.

I shall be paid a princely sum I know without a doubt,
Should I ever be struck by Mohamed Ali in a fistic bout.
Or if it should happen on the Inter-State Eighty One
That my dog sled team and I should by a bus be overrun.

Now these things will make me rich yes indeed.
But right now this is not my one and only need.
I want to get even with that Computer so smart.
It thinks it has outsmarted me in whole or in part.

But I have a plan that I know it will surely succeed,
To take away that one thing that all computers need.
Take away this and they lose their respect and control
Yes, I mean to steal each and every punch card Hole!!

Edward Stewart

SECRET LOVE

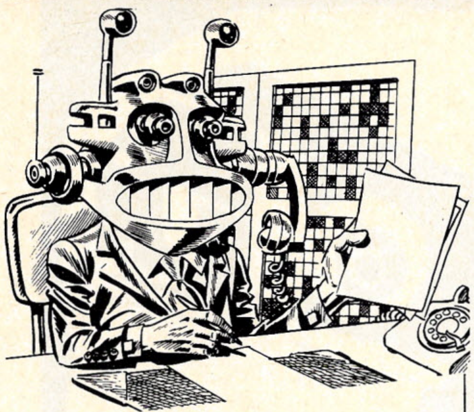
Computers have no heart people very often say.
Nonsense! The one I worked with is warm, bright and gay.
It is smart and fast, friendly too, I may wish to add.
It is the best friend that I have ever had.

I have talked to it for hours, when the others are all gone.
Talked to it thru the night almost to the dawn.
My machine won't tell a soul, confides in me alone.
I tell it secret things, like the brand of my cologne.

We have talked for many hours, far into the night,
Argued too, about fishing, when and where they bite.
Some at the office now start to give me long odd looks,
When I start to read to it from out of my dirty books.

They have put me away now, in a brand new home,
That gives me more time to write and draw up my own
tome.
So it don't matter so very much what they have to say.
I would write much more but they took my crayons away.

creative computing brings you its best!



Creative Computing Magazine

The #1 computer applications magazine helps you get the most out of your computer, be it a micro, mini or timesharing system. Plenty of practical applications in education, small business management, household management, recreation and games, and building control. Programming techniques, for those writing their own, include sort techniques, recursion, and file structures. Complete listings and sample runs of games, investment analysis, music synthesis, artificial intelligence programs and much more. Articles, in-depth book and resource reviews, hardware and software evaluations and even some fiction and foolishness. "But the best thing about *Creative Computing* is its feisty, friendly attitude" — *The Village Voice.* Subscriptions, USA 1 year $8 [1A], 3 years $21 [1B]; foreign 1 year $12 [1G], 3 years $33 [1H]. Sample copy $2 [1C]. Volume 1 and 2 issues have been edited into two big 324-page books. *American Vocational Journal* said of Volume 1, "This book is the 'Whole Earth Catalog' of computers." $8.95 [6A]. Volume 2 continues in the same tradition, "Non-technical in approach, its pages are filled with information, articles, games, and activities. Fun layout." — *American Libraries.* $8.95 [6B]. Also available is a special package of all the single issues in Volume 3 for only $8 [1N].

Some Common BASIC Programs

Adam Osborne. An ideal workbook to accompany a BASIC programming course or for one's own use. Contains 76 general purpose, practical BASIC programs with complete listings. 192 pp. $7.50 [7M].

Game Playing With Computers

Donald Spencer. A good mind-exercising resource. Contains over 70 games, puzzles and recreations for the computer. The games, in both BASIC and FORTRAN, have detailed descriptions, flowcharts and output. Also includes a history of game-playing machines and lots of "how-to" write your own games. 320 pp. hardbound $16.95 [8S].

BASIC Programming

Kemeny and *Kurtz.* An excellent gradual introduction to computer programming in BASIC with lots of samples ranging from simple to complex. Rated "The best text on BASIC on almost all counts." 150 pp. hardbound $8.50 [7E].

Byte Magazine

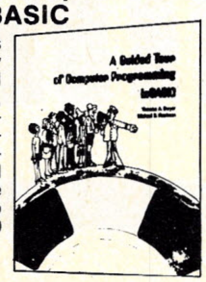
If you're considering your own personal computer, then you need *Byte.* With an emphasis on hardware, *Byte* provides a wealth of information on how to get started at an affordable price. Theory, opinion, how-to articles. Subscriptions, USA 1 year $12 [2A], 3 years $32 [2B]; foreign 1 year $25 [2F]. First issues of this now classic publication have been edited into *The Best of Byte, Volume 1.* 386 pp. $11.95 [6F].

Microcomputer Design

Donald P. Martin. This book is well-suited for the engineer who's designing microprocessors into his company's products. Not just block diagrams or vague theory, but dozens of practical circuits with schematics for CPUs based on 8008 chips. Includes interfacing to A/D, D/A, LED digits, UARTs, teletypewriters. Over 400 pp $14.95 [9P].

A Guided Tour of Computer Programming In BASIC

Dyer and *Kaufman.* This book tops all introductory texts on BASIC. Filled with detail and examples, it includes sample programs for many simulations, several games, reservations systems and payroll. Aimed at the novice, but of value to everyone. 156 pp. $4.80 [8L].



For faster service call toll free

800-631-8112

In NJ call 201-540-0445

Artist And Computer

"Get yourselves a copy of this book if you enjoy feeding your mind a diet of tantalizing high-impact information." — *San Francisco Review of Books.* In this book, 35 artists present a multitude of computer uses and the very latest techniques in computer-generated art. 132 pp. $4.95 softbound, [6D], $10 hardbound [6E].

BASIC Computer Games: Microcomputer Edition

David H. Ahl. An anthology of 101 games and simulations. Everything from Acey-Deucey to Yahtzee, all in BASIC. The only book of its kind with complete listings, sample runs and descriptive write-up of each game. 248 pp. $7.50 [6C].

Creative Computing Catalog

FREE 16-page reference to computer-related products not readily available in commercial stores. Lists books, magazines, T-shirts, game, binary dice and more [5A

The Colossal Computer Cartoon Book

Sit back and relax. Take a break with the best collection of computer cartoons ever. Popular gift item, too. 128 pp. $4.95 [6G].

Problem Solving With The Computer

Ted Sage. Used in conjunction with the traditional high school math curriculum, this book stresses problem analysis in algebra and geometry. This is the most widely adopted text in computer mathematics. 244 pp. $6.95 [8J].

Computer Rage

Fun and educational new board game based on a large-scale multi-processing computer system. The object is to move your three programs from input to output. Moves are determined by the roll of three binary dice representing bits. Hazards include priority interrupts, program bugs, decision symbols, power failures and restricted input and output channels. Notes included for adapting game for school instruction. Great gift item. Ages 8-adult, 2-4 players. $8.95 [6H].

Getting Involved With Your Own Computer

Solomon and *Viet.* One of the first books on microcomputers that requires no previous knowledge of electronics or computer programming. Tells you where to find information, explains basic concepts and summarizes existing systems. Good place for the neophyte to begin. 216 pp. $5.95 [9N].

Fun & Games With The Computer

Ted Sage. "This book is designed as a text for a one-semester course in computer programming using the BASIC language. The programs used as illustrations and exercises are games rather than mathematical algorithms, in order to make the book appealing and accessible to more students. The text is well written, with many excellent sample programs. Highly recommended." — *The Mathematics Teacher.* 351 pp. $6.95 [8B].

An Introduction to Microcomputers

Adam Osborne. Volume 0, "The Beginner's Book" covers microcomputer system components, how they work together; number systems, the basics of programming, and putting it all together in a system of your own. 264 pp. $7.50 [9T]. Volume 1, Basic Concepts, also assumes no prior knowledge of computers. It covers basic principles, binary arithmetic, the microprocessor CPU, I/O logic, memory organization and programming. 264 pp. $7.50 [9K]. Volume 2 (2nd Edition), Some Real Products, covers 20 actual microprocessors in considerable detail including timing diagrams, instruction sets, and interrupts. 760 pp. $15 [9L].

creative computing Attn: Cheryl P.O. Box 789M, Morristown, NJ 07960 USA

Please send me the following:

Quan.	Cat.	Description	Price

Books shipping charge
$1 USA, $2 Foreign ______
NJ residents add 5% Sales tax ______
TOTAL ______

Name ______
Address ______
City ______ State ______ Zip ______
☐ Cash, check, M.O. enclosed
☐ Bank Americard/VISA
☐ Master Charge
Card No. ______
Expiration date ______

Richard Grant

Illustrated by Richard von Magnus

I wasn't doing much that afternoon, just twiddling my memory banks and staring out the window, when she walked into my office. All five foot eleven of her. She was wearing a black silk blouse that was open down the center to the navel, a black leather skirt slit up the side, black shiny boots, and a wide-brim straw hat that hid her eyes and made her seem taller. A thin brown cigarette protruded from her fingers, dropping ashes silently on my rug.

It was a swell outfit. The only thing missing was a panther on a leash. She'd probably left it double-parked outside.

I said: "Take a chair, sweetheart, you've already taken my heart."

She whirled around, startled, and stared at me. Her jaw dropped. "You? You're Sam Data, the detective?"

"That's what the sign on the door says."

She smiled, a thin hard smile. Then she threw her head back and laughed. Between gasps for air she got out, "It's...it's just that I was expecting... ."

"Yeah, I know, you were expecting Humphrey Bogart. I can talk like him, if that's any help," I said, switching my voice-syntax box over to Bogart. It limited my vocabulary a little, but I didn't think it would matter.

She sat down on the hardwood chair I provide for my clients, and hitched her hat back. That gave me a look at a pair of smoke-filled green eyes. "A computer private detective." she said, shaking her head. "No wonder you don't make house calls."

"Figured that out all by yourself, did you."

She laughed again. "Bogart, all right. Down to a T. This is just too much. Who writes your dialog?"

I let that one go. Business hadn't been too good lately, and I heard a rumor that if it didn't pick up I might soon be recording parking violations. I needed a client, so I turned on all the Sam Data charm. "My programmer was a great fan of hard-boiled detective fiction. Some of it rubbed off. Now did you come on business, or are you just up here to admire the view?"

Her face straightened out, and she looked disdainly around my office. There wasn't much to look at. A couple of chairs, a table with a few dusty magazines, and four empty walls badly in need of a paint job. Outside, the steady stream of traffic up and down Sunset sounded like a river heard from a long way off.

"Business," she said finally. "I, I don't know quite how to say this."

"Try moving your lips."

She sneered at me, dropped her cigarette on the rug, and ground it to pieces with a six-inch heel. "It's about my husband, Robert. Robert Barkley."

"What about him?" I said.

She smiled nervously. "Well, he's dead."

I ran that one through, then said, "Uh-huh. Since June 6th. A .38 slug in the head. One shot fired at, or about, 3 a.m. Ballistics match it to a .38 automatic which just happened to be found in the deceased's hand."

She stood up, her face pale. "But...how... ."

I ignored her and went on. "Could be a suicide. He left a typewritten note, but there's one problem. He forgot to sign it. The note contradicted his will, and left everything—and I do mean everything—to his nephew. Bad break for the wife, wasn't it, Mrs. Barkley? Or should I call you Susan?"

She was still standing, the crown of her hat just about scraping the ceiling. "Do you mind telling me how you know all this?"

"No, I don't mind. I have some friends down at police headquarters. I've helped them out on a few cases, and in return they let me keep an open channel with their computers. Now let me ask you a question, just what is it you expect me to do for you?"

She sat down again, and lit another cigarette. "This channel with the police, is it two-way? I mean do they hear what I'm saying?"

"Not unless I want them to. And right now I don't want them to."

She looked a little skeptical, so I said, "Listen Mrs. Barkley, if I didn't respect a client's confidence I wouldn't be in business for a minute. The police know that. They let me keep the open channel because when things are slow, I do some work for them, that's all. I can shut it off, or make it one-way any time I want to. Understand?"

The straw hat nodded up and down. "I guess it doesn't matter if the police know anyway," she said, taking a large drag on her cigarette. "Of course the reason I need you is to prove that the suicide note is a phony. I know it is. Robert and I were, well, we weren't on the best of terms, that's no secret, but he wouldn't have cut me off like that. And he wasn't the type to commit suicide. He was too much of an egoist for that."

"Uh-huh," I said, and pressed a few buttons. In ten seconds the piece of slick paper was ready. I fed it out of my bottom slot so she would have to bend over to grab it. That gave me a quick flash of her legs. They were nice legs.

"What's this?" she said.

"It's a photostat of the suicide note."

She looked up, impressed. I went on, "It was typed on an IBM Selectric—the one in your husband's office—on a carbon ribbon. There were standard business letters on the ribbon before the note, and nothing after it, so it was the last thing typed. I'm studying your husband's other correspondence now. And from the looks of it, I'd have to agree with you. He didn't write that note."

"What?" she practically yelled. "How can you tell already?"

"Patterns. They're as evident in writing as they are in speech, of course, not to the average person though. By examining a large sampling of someone's writings, and giving mathematical configurations to certain words, a definite pattern emerges. The pattern in the suicide note is not your husband's."

"I've never heard of this before."

"Yeah, well actually it's not totally accepted by the law enforcement community yet. It's just something I've worked out—one of my many services."

"Yeah," she said. "Last year's jokes and tomorrow's crime-detection techniques."

I let that one go by, too. "Funny thing about these word patterns. I can tell who did write that suicide note if I have some samples of their writing."

"Yeah?" she said nervously.

"Yeah." I wasn't sure, but I have it a try. "You know as well as I do, Mrs. Barkley, that an unsigned suicide note carries no weight with the law in this state. You get the whole inheritance just like the original will says. But the note was clever, I'll give you that. It threw suspicion on the nephew, without causing you any great danger. In order for it to work, though, you had to pretend to be concerned about it, so you decided to hire a private detective. That was your only mistake. The suicide was easy enough to fake. You probably just shot him in his sleep, or..."

She stood up fast, knocking the chair over behind her. "You can't prove that..."

"Don't kid yourself, sister. The cops are already on their way over. I put out a call to them five minutes ago. And don't bother trying the doors. They're on an electronic lock system, and I already switched them on."

Her green eyes flicked to the door, then back to me, spitting daggers. But her voice literally purred. "You don't want to tell the cops anything," she said walking towards me. She put her fingers out and stroked my data-control center. Her fingernails were painted black. "We could be good for each other, Sam. You're right about the will—all that money is mine now. Or should I say ours. I can have you moved out of this shabby office, we can go to Rio, or we can..."

"Can it. You're taking the fall, kid. Oh sure, we might be good together for a few weeks, maybe even a month. But then one morning I go to recharge and find my power-pack unplugged."

"You've thought of everything, haven't you," she said, her voice taking on a hard edge. "Except this." I was looking into the barrel of a small automatic.

"I told you—the police are on their way." I was still using Bogart's voice, but it sounded funny now.

She laughed. "Let'em come. What's the penalty for shooting a computer?" There was a tiny spark, the gun jumped in her hand, and one of my electronic eyes shattered. There were two more hard slaps, and then the sound of ripping metal. Damage reports flooded in. I put out an all-points bulletin for every police car to come to my address. It was a police code I wasn't even supposed to know, let alone use, but for once I was glad I didn't always play it straight with the police.

Three more shots rang out in the small room. Blue smoke hung in the air.

She hadn't hit anything vital yet, but it was only a matter of time. I had to take a chance. Over by the door, about ten feet behind her, I had a small speaker I used to talk to the mailman with. I swtiched it on, and suddenly Lee Marvin was saying, "All right sister, hands in the air, slowly, and don't turn around or you've had it."

She jumped about a foot.

"Didn't know Sam had a partner, did you?" Marvin said. "Only this partner's flesh and blood."

It didn't sound like flesh and blood. The speaker had a static like noise to it, but it was good enough to fool her. She raised her arms in the air.

"Sam, open a drawer."

I threw one of my electronic storage drawers open.

"Drop the gun in it, sweetheart."

She did as she was told, and I slammed the drawer shut. Maybe the Pittsburgh Steeler Front Four could have opened it again, but she couldn't.

Slowly, she turned those beautiful green eyes towards the door. "But...where...you...you tricked me!"

"That's right, honey," Lee Marvin said from my main voice box.

She whirled around on me. "Okay. So you got my gun, so what. No jury in the world is going to believe all that word pattern mumbo-jumbo. You haven't got anything on me."

"You're right about the word patterns. That stuff is straight from the comic books. But everything you've done and said in this office has been tape-recorded and video-taped. Should be enough to give the cops something to think about. You see, that chair you were sitting on, when you weren't pumping bullets into me, is an electrically-wired lie detector. My own invention. I got a hunch about you from some of the readings I was getting, and I decided to play it."

Her shoulders sagged. "And I fell for it."

"From twenty stories. But don't feel too bad. With your legs, you should get off pretty easy." I opened another drawer, the one where I keep the office bottle, and said, "Have a drink."

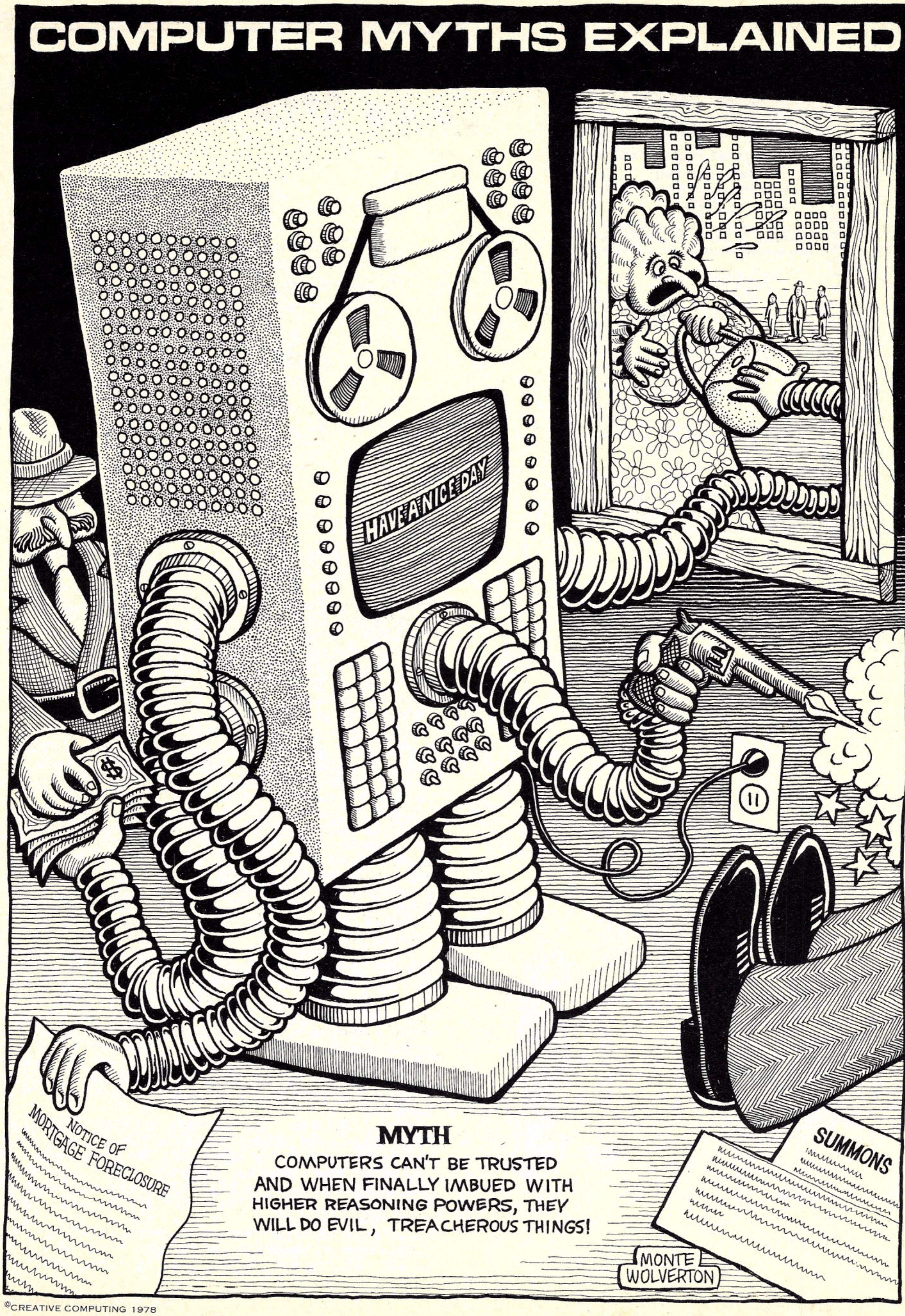
In the distance, the wail of police sirens cut through the noise of the traffic below. She took the bottle and the dirty glass next to it, and poured herself a healthy slug. She threw it down with one quick jerk of her head, and poured another. Her green eyes lifted to one of my unshattered electronic ones.

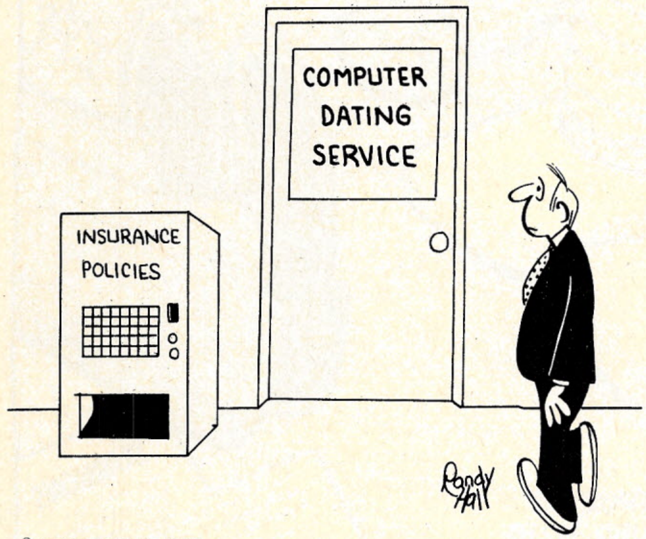
"You know Sam, I wasn't on that lie detector thing of yours when I was stroking you, but I did feel something. Really, I did. You know, you're kind of cute."

"Yeah," I said.

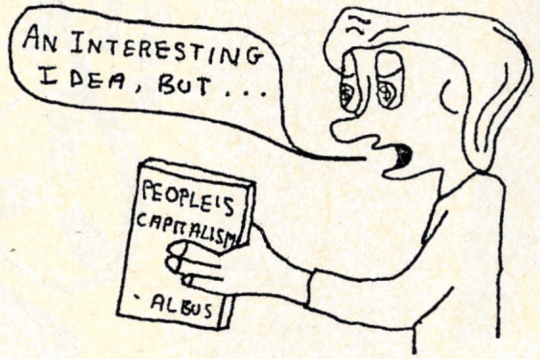
"Maybe I'll get off easy. We could start off fresh when I get out." She walked over, and ran her arms up my data banks, carefully avoiding the holes where her bullets had gone in.

"Sure kid," I said, as I unlocked the office for the two uniformed policemen standing. "I'll be waiting for you." ■


COMPUTER MYTHS EXPLAINED
HAVE A NICE DAY
NOTICE OF MORTGAGE FORECLOSURE
MYTH
COMPUTERS CAN'T BE TRUSTED AND WHEN FINALLY IMBUED WITH HIGHER REASONING POWERS, THEY WILL DO EVIL, TREACHEROUS THINGS!
SUMMONS
MONTE WOLVERTON



... reviews.



People's Capitalism: The Economics of the Robot Revolution. James Albus. New World Books, College Park, MD. 157 pp., paperback. 1976.
The First West Coast Computer Faire Conference Proceedings. Jim Warren, Editor. Computer Faire, Box 1579, Palo Alto, CA 94302. 336 pp., softbound. $12. 1977.
Computer Concepts for Management. Two cassette tapes, 80-page loose-leaf manual. $150. 1976.
Developing User-Oriented Systems. Two cassette tapes, looseleaf manual. $175. 1975. GROUP/3, 21050 Vanowen Street, Canoga Park, CA 91304.

Albus points out that most people in our employee society are wage slaves — no wage or salary, no income. As automation reduces the number of real, worthwhile jobs, their place is taken by often intrinsically worthless make-work jobs simply because the wage is the primary method of income distribution.

People's Capitalism advocates replacing or supplementing wages with income from a National Mutual Fund, established by Congress and funded through loans from the Federal Reserve System. The NMF provides capital to enable businesses and factories to modernize and increase production. Thus the people would be the capitalists funding industry and the dividends of the NMF would be distributed to the people, eventually making them largely independent of wages as a source of income.

It's an interesting idea, if one believes in the fanciful notion that the United States is a democracy. However, Albus does not adequately deal with the already massive problem of what 225 million people are going to do with leisure time coming out of their ears, the distinction between labor and work, and the tendency for automation to centralize power and engender a technological elite. Also, it seems to me that distribution of the NMF dividend would undoubtedly require assignment of a universal product code, er, excuse me, identification number to individuals. When that time comes, I'm going to join the neo-Luddites.

The First West Coast Computer Faire Conference Proceedings is very difficult to describe. It is a rich and fertile compendium of papers covering topics that run the gamut of everything even remotely concerning home computing. Something for everyone, etc. If you want to know what personal computing is all about, thumb through the *Proceedings.* You'll come away wiser for it.

CCM is a very elementary introduction to Data Processing for Management type people who have little or no prior training in DP. The manual might loosely be termed cartoons,

. reviews . . .

accompanied by the professionally scripted and produced cassettes that don't shove facts at you more quickly than they can be digested. This course is aimed at management above the DPM level to give them some idea as to what goes on in the DP department, hopefully encouraging them to ask questions now that they have some idea how. This is a real need in some businesses and institutions. The busy executive can go through this course in an evening of two or three hours and then pass it on to someone else in the company. I feel that this one is a little over-priced.

DUOS is not simply a course, but a complete system. What is being presented here is both a development system *(DUOS)* and a general course on how to use any development system. The cassettes provide a structured presentation of the system, interspersed with nineteen question and answer breaks. A sample system (a pension plan) is taken from beginning to end, showing the proper use of *DUOS*. *DUOS* is a set of procedures and forms that make up a flexible development system. This is aimed at Data Processing Managers and System Analysts. Since an entire methodology and even reproducible forms to get you started are included here, I feel that this price is reasonable.

SISTM, the Society for the Interdisciplinary Study of The Mind, has published Vol. 1 No. 1 of their journal, the *SISTM Quarterly*. Articles include "Research at Yale in Natural Language Processing," "Consciousness as a Workspace," and "The Problem of Context in Neuropsychology." Information and membership ($10) is available from Dr. Martin Ringle, Editor, Box 693, Phillipsport, NY 12769.

Travels in Computerland, or Incompatibilities and Interfaces. Ben Ross Schneider, Jr., Addison-Wesley Publishing Co., Reading, Mass. 244 pp. softbound. 1974.

Ben Schneider is an English professor who had a vision. As a theater historian, he felt it would be useful if the huge reference work *The London Stage, 1660-1800* could be contained in a computerized data base, to assist scholars in their research. Not knowing much of what would be involved, he jumped at the opportunity to direct a project to achieve his goal.

Travels in Computerland is an entertaining saga of his misadventures; the subtitle, *Incompatibilities and Interfaces,* effectively summarizes Dr. Schneider's quest. The 8,000 pages of text were edited by graduate students scattered around the U.S., keyed by typists in Hong Kong, optically scanned and converted to computer tape in London, and finally processed by the computer at Lawrence University in Wisconsin, which was to be the home of the London Stage Information Bank. In the course of all this, Dr. Schneider learned of the complexities of managing a data-processing project, of dealing with computer companies which were unsure of just what their machines could do, of coping with customs agents in London and at home, and of the problems of raising funds for humanities research. That the Information Bank became a reality at all is probably due to his naivete in all these areas, for had he known what was facing him, he may well have considered his task impossible from the start.

As an outsider to "Computerland," Dr. Schneider is able to present simple, clear explanations of the mechanics of information retrieval, along with some of the theory of cybernetics. And finally, he provides a compelling demonstration of the valuable role which can be played by computers in the humanities.

Thomas A. Gutnick
Arlington, VA

Computer Dictionary. Donald D. Spencer. Camelot Publishing Company, Ormond Beach, Florida. 160 pp. $9.95 ($5.95 paper). 1977.

On the thought that "a rose is a rose is a rose" might translate into the realm of ADP as "a word is a word is a word," I began my review of this volume by looking up "word." I was informed tersely and clearly (but not necessarily precisely or unambiguously) that a word is "a group of bits, characters, or bytes considered as an entity and capable of being stored in one storage location. See KEYBOARD." Cross-references? Good idea. But the definition of keyboard doesn't have anything to do with or even reference "word." Sloppy, but not fatal.

I decided to let myself be led about by the cross-references and looked up "microcomputer." I was led through "home computer," "personal computer," "computer kit," and "microprocessor," all of whose definitions taken together give a good feel for what the subject is all about. If one is curious about the acronyms used in the definitions, they are all available in the dictionary itself. As a matter of fact, this book can be very useful to its intended audience in sorting out the alphabet soup that tends to creep even into articles written "for the layperson." Included are such acronyms as ASCII, ASR, KSR, MOS, LSI, and even GIGO. Look up ROM or PROM and you see the real words all laid out. You also find a closed loop — poor debugging techniques, I presume — when PROM says "See PROM" as its cross-reference.

Some definitions are indeed definitions, but are wrong. The ILLIAC IV is no longer in Illinois, but several years ago was moved to the Ames Research Labs in California. IBM is not the only manufacturer of a 96-column punch card. These are simply instances of information being overtaken by events. Not so with the definition of multiprocessing which is too much directed at the program rather than at the hardware technique involved. It *should* say that multiprocessing is the simultaneous execution of two or more sequences of instructions by multiple central processing units under common control. And, for the record, the real definition of "kludge" is "an ill-assorted collection of poorly matching parts, loosely fit together to form a distressing whole." But its inclusion is indicative of Mr. Spencer's sense of humor — where else could you find "gulp" defined? (It's a small group of bytes.)

Mr. Spencer's book is described as an authoritative reference book designed to assist students, technical readers, and non-specialists whose work is to some extent affected by the computer. It is a good volume, but not necessarily an exhaustive or precise one. It can serve as a useful reference for the outright novice and as a memory jogger, but by no means the ultimate or complete authority, for the technically-oriented. A library could do much worse than to have it on the shelf. But, hopefully some future revision will make the next edition even better.

Deanna J. Dragunas
Wetumpka, AL

Computer Programming Handbook. Peter R. Stark. TAB Books, Blue Ridge Summit, PA 17214. 506 pp. $8.95. 1975.

This is a great book . . . for the right person. Essentially, it is an introduction to programming, first in machine language, then in assembly language and finally in a higher-level language (Fortran). But it also contains a significant section on programming techniques that, although most of the examples are worked out in assembly language, covers a broad variety of topics beyond what one would ordinarily encounter in an introduction to programming and covers them very well. And it contains short sections on the history of computers, computer applications, computer organization and flowcharting as well. There's a lot of stuff here, and by and large it's very well done.

I found this book very well written and easy to follow but that can be misleading. I think I knew most of the things that Stark discusses before I read his discussion. Still, he has a good knack for writing clearly and I don't think that I am wrong in saying that his explanations would work well, even for people for whom what he is explaining is new. If you have a modicum of mathematical ability (you didn't flunk highschool algebra) you should find this book fairly easy to follow. You will have to

think, of course, but there are lots of worked-out examples to help you and Stark tells us that they have been checked out on a computer so they shouldn't be misleading. His explanations of algorithms are unusually clear. I particularly liked his brief history of the computer, his explanation of what a subroutine is, what a function is and a number of nice little points like the clear way he explains Newton's algorithm for finding square roots. The machine and assembly languages are simplified versions and the higher-level language is Fortran. The index is good.

The book has fairly broad scope and it is probably best for somebody who knows a bit about programming and wants to read clear explanations of things that he or she has not yet learned. I suspect it's not really a good first book on computers although it might be that for the right (smart) reader. It's a good second book though.

Peter Kugel
Chestnut Hill, MA

Eight Statement PL/C (PL/ZERO) plus PL/ONE. Michael Kennedy and Martin B. Solomon. Prentice-Hall. 512 pp., paperback. $10.50. 1972.

Eight Statement PL/C is designed to teach a beginner the fundamentals of programming in PL/I. True to its title, the book teaches an eight-statement subset which is enough to get any beginner off to a flying start in the language. The PROCEDURE, END, DECLARE... FLOAT, GET LIST, PUT SKIP LIST, IF ... THEN, arithmetic assignment, and GOTO statements are covered very thoroughly in a 100-page section. The second section introduces a wider subset of PL/I; it includes fixed binary and character type declarations, DO looping, input/output editing, and internal procedures. Part three contains short chapters on selected PL/I topics such as data-directed input/output, conditions (e.g. ENDFILE, ZERODIVIDE, etc.), debugging aids, complex numbers, PICTUREs, record I/O, list processing, and the more advanced storage allocation techniques for static, controlled, and based variables.

By now, perhaps you may tell that the book advances far beyond the basic "eight-statement" language. Nevertheless, it *is* a book for beginners. There are appendices on the punched card, simple keypunch operating instructions (excellent for the beginner), algorithms and flowcharting, computer system basics, numerical inacuracies (floating point error), a short history of who made which computer when, and a summary of the idiosyncracies of each of the IBM PL/I compilers as well as Cornell's PL/C. There is a variety of laboratory problems provided and each is cross-referenced by the sections of the book that should be covered prior to attempting it.

Eight Statement PL/C is full of good illustrative examples and has been constructed so that immediately after a concept has been introduced, an example appears. Then the authors present a complete explanation of the feature (with appropriate references to the sample, of course). Not only does this technique stimulate asking questions about examples before being given "the answer"(which study experts claim is a good way to absorb something technical), but it grabs and holds the reader's interest much better than would a series of explanatory paragraphs *followed by* a few examples.

The only fault that could be found with the book is that while the examples are well structured and clear, the authors do not explicitly encourage what is today called "structured"programming. For this reason, in 1977 they published *Structured PL/ZERO plus PL/ONE* (Prentice-Hall, $11.50) in order to satisfy this need. The second book differs enough from the first that the earlier one is still in print. *Structured PL/ZERO* contains a slightly different set of eight basic statements to help the beginner with structure and it includes some philosophical comments on the manner of programming clearly scattered through the book. I personally prefer the earlier book for its practical approach to the programming basics, but either could be used in a first course in computer programming using PL/I as the teaching language. Both for self-study and in the classroom, each book is excellent and worth every cent of its price.

Brian N. Hess
Western Springs, IL

From: Kemeny, John G. *Man And The Computer.* New York: Charles Scribner's Sons, 1972. pages 80-81.

"I consider it imperative for the benefit of mankind that during the next decade computers become freely available at all colleges and universities in the United States and that most students before graduating acquire a good understanding of their use. Only if we manage to bring up a computer-educated generation will society have modern computers fully available to solve its serious problems. While computers alone cannot solve the problems of society, these problems are too complex to be solved without highly sophisticated use of computers. I see three major bottlenecks that must be removed if this goal is to be achieved.

"First, most university computation centers are still research-oriented. They are typically operated in a batch-processing mode with priorities given to a very small number of users who need a great deal of time. The philosophy of university computation centers must be changed.

"Second, college administrations do not yet appreciate the immense favorable impact that a good educational computation center can have on their institution. I would like to propose that by 1980 no college or university should be given full accreditation unless computer services are freely available to all students. Use of the computation center must be considered the exact analogue of use of the library.

"Finally, the implementation of this program for millions of students will take a great deal of money..."

"HANDS-ON" AND FAST TURNAROUND -A MUST FOR A SUCCESSFUL PROGRAM

by THOMAS J. CASHMAN and GARY B. SHELLY

INTRODUCTION

A continuing controversy in schools teaching computer programming concerns the need for "hands-on" experience in the data processing curriculum. Is "hands-on" experience necessary in teaching data processing and computer programming? What about turnaround time? In many schools, the students are not allowed to have any contact with the computer. In addition, students may receive only one or two "runs" a day for programs written for class assignments. The authors feel strongly that students must have actual experience on the computer and that the turnaround time for student assignments must be as fast as the computer center can process, not what might be expected in industry.

LOSS OF COMPUTER

The scenario at many schools runs something like this: In the beginning, instructors have gone to the administration requesting a computer for instructional purposes, pointing out the vocational opportunities in data processing. The administration has enthusiastically agreed with them and authorized the leasing or purchase of hardware, on the grounds that the school should have a strong instruc-

Reprinted with permission from *The Compiler*, a newsletter for business data processing educators. Copyright 1975 by Anaheim Publishing Co., 1120 East Ash, Fullerton, CA 92631.

tional program in data processing. When the computer arrives and the data processing program begins, students have the opportunity to operate the computer, experiment with the computer, and generally learn and have fun in their data processing classes. As a result, student enthusiasm runs high.

With the passing of time, however, an "administrator" realizes that none of the grading, scheduling, payroll, or other applications which could be on the computer are being processed. Therefore, he convinces the powers-to-be that the administration should take over the data center and process administration applications. Control of the data center then falls into the hands of the administration, who will determine when programs for students will be processed.

A data processing manager is hired. Students can't be allowed in the data center; it is reasoned that "all those students milling around is no way to run a 'shop'."

As more and more processing for the administration is placed on the computer, the instructional program becomes less and less important. Student programs are run once or twice a day instead of the six or seven times a day as previously. Instead of receiving a class assignment back in 15 minutes, the student must wait four or five hours. When complaints are raised by the instructional staff they are told "You're lucky to get 4- or 5-hour turnaround time—in industry 24-hour turnaround time is common." The instructors reply "This is **not industry**, this is an educational/academic environment in which the important objectives are to motivate, to stimulate, and to develop a spirit of inquiry and research." Such statements commonly fall in deaf ears. "That's not the way it's done in industry" is the continued reply.

Student enthusiasm for data processing wanes, students are learning less because of no access to the machine, enrollment goes down because of a lack of student interest, the administration points out that the only justification for the computer is now administration processing because of low student enrollment, and, as a result, the initial justification for the computer, that is, the instruction in data processing, takes a back seat to every other type of processing on the computer.

If this scenario sounds familiar, do not despair. Hundreds of schools in the United States and Canada are facing the same problem. The surest way to weaken and destroy a viable data processing educational program is to take control of the computing facilities away from the instructional staff and place it in the hands of "professional" data processing managers whose job depends upon the processing of administrative applications.

ARGUMENT OF DATA PROCESSING MANAGERS

As previously pointed out, the argument of these "professional" data processing managers runs along two lines. First, they argue that programmers in industry do not have access to the computer and they get along fine. Secondly, they point out that in industry, many programmers only get one test run per day. They conclude that this should work for students also.

The flaw in both these arguments is the fact that students in a data processing instructional program are not programmers in industry. An extremely important element in any successful educational program is the student MOTIVATION and enthusiasm for learning. Although it may be difficult to quantify, it has been the authors' experience that enthusiasm for learning is almost directly related to access to a computer when teaching data processing. The more the student is exposed to the computer and the better access he has to it, the more he will learn and the more he will want to learn.

It has been observed numerous times that when students operate in a "hands-on" environment, for many their learning far exceeds that which is expected in the class, their knowledge gained quickly exceeds that of their instructor, and many students become software system experts! What a great compliment to the computer as a motivating device for learning. What a shame to prevent these learning experiences from occurring! The authors have yet to contact or observe a school that operated on a closed-shop basis that produced operating system experts—experts in DOS, OS, etc.

It seems ironic that instructors with many years of experience in the classroom successfully teaching data processing and computer programming are told by administrators or data processing managers (who, by the way, have never taught a class in computer programming) that "hands-on" experience is not necessary!

As noted by Malcolm G. Lane from West Virginia University, "Experience has shown that the 'hands-on' approach is an excellent way for students to learn meaningfully the operating system principles being studied in the course and to observe the effects of well- and poorly-designed systems. One reason is that computer science **students gain an enthusiasm in the course when the 'hands-on' approach is used; this enthusiasm is more difficult to provoke when other methods are used.**" [1]

MUST EXPERIENCE COMPUTER ACTIVITY

In addition, it is extremely difficult to teach programming without students experiencing the satisfaction, and even thrill, of seeing their program work. Teaching programming without access to the machine is like teaching chemistry without access to a chemistry lab or teaching literature without reading the book. Certainly an instructor in chemistry can describe what happens when a match is held to a test tube of oxygen, but when the student does this and sees the combustion, it is something which he will not soon forget. An English teacher can talk about the **Odyssey** and describe the travels of Odysseus, but without reading the book the student never experiences the learning process.

"Teaching programming without access to the machine is like teaching chemistry without access to a chemistry lab or teaching literature without reading the book."

The same is true of computer programming. Without experiencing the lights flashing, the cards being read, and the program and output being printed, the student misses a critical part of the learning process. In addition, with access to the computer, the student can experiment with features of the hardware and operating system which would not be possible without access to the machine.

EDUCATORS RECOGNIZE THE NEED

At Scott Community College in Bettendorf, Iowa, a "continuous effort is made to provide the type of training needed by students and required by business." As a large part of their quality program, "students have almost unlimited hands-on time in a realistic multiprogramming environment."[2] They have recognized the undeniable requirement of hands-on experience for programming students when learning computer programming.

Undeniably, this is a difficult argument to present to an administration which has its "professional" data processing manager stressing to it that this is not the way things are done in industry. However, it appears that a number of schools at this time are allowing hands-on experience for their students. Professor Marjorie M. Leeson, Delta College, University Center, Michigan, conducted a survey of 72 community colleges and found that 54 of these schools allow some type of hands-on for their programming students.[3] Although this is encouraging, the authors have found that the trend is away from hands-on as more applications of an administrative nature are put on the computer.

Thus, it is very important that administrators be made aware of the importance of hands-on experience for the programming students. Perhaps an effective method to illustrate the point is to invite administrators to a Saturday class in which they will be taught enough of a language to write two programs. On the first program, they are allowed to enter the computer room, run the machine, see their program being produced on the computer, and get immediate turnaround. On the second program, they must wait two hours for their first run and two more hours for their second run. In addition to probably not completing the assignment, these administrators, as are the students, will be very frustrated.

CONCLUSION

If your school has not yet reached the point where the computing facilities have been taken out of the instructional program's hands, fight to not let this happen. Where it has happened, you owe a duty to yourself and your students to vigorously fight to regain access to the machine which is so vital to a successful data processing program. ■

1 Lane, Malcolm G., Department of Statistics and Computer Science, West Virginia University, Morgantown, West Virginia, "A Hands-On Approach to Teaching Systems Programming," **SIGCSE Bulletin**, February 1975, Volume 7, Number 1.

2 Stone, L.E., Director - Data Processing, Scott Community College, Bettendorf, Iowa, "DP Training—Meeting the **Real** Needs of Business."

3 Leeson, Marjorie M., Delta College, University Center, Michigan, Summary of Results, Data Processing Questionnaire.

The square r

It's 1.99999. Those of us who have used either computers or desk calculators for a long time are familiar with the problem. Most of the time everything seems to come out with good answers, but once in a while something obvious just isn't right.

Oh where has my integer gone?

There are three sources of error inherent in the computing process which can cause these problems. The first is word size. No machine made is capable of representing all the digits in an irrational fraction. It is possible to do exact computations by carrying numerator and denominator separately and never dividing, but this is a trick used by mathematicians and numerical analysts and rarely finds its way into everyday computing.

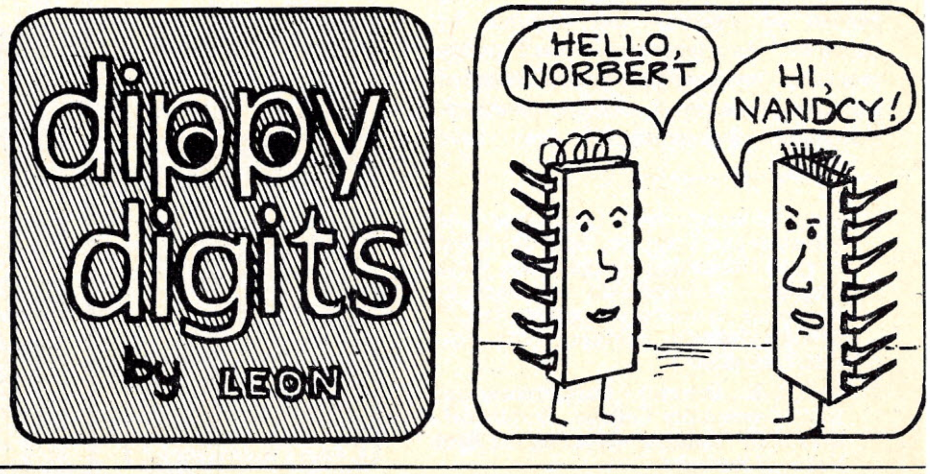
The second source of error is that all mathematical routines used to find functions such as square roots are approximations. To build a program which calculates the most accurate value of the function that can be represented in the machine (in error by at most one least significant bit) would require more computer time and storage space than the result is worth.

There is a continuous tradeoff among speed, accuracy and size. Each designer has mixed these in his own proportions, and his choice may not coincide with yours. That last single bit error, though, is a result of having a finite word size, and cannot be improxed. This means that the square root of 4 may come out 1.99999. Although this may insult some users who feel that "the computer ought to be able to get it right" and who know that the square root of 4 is 2, it is an inevitable consequence of working with real computers.

Trying to make matters better would almost certainly make them worse. It is possible to design a function program that would give the "right" answer for some value (e. g., square root of 4 is 2.00000). This would most likely cause the margin of error for all other values to be greater.

Round off, one-two

It is also possible to build in an automatic round-off to a function. A small amount could be added to the answer after it was caculated, and a round-off attempted. This would give 2 as the square root of 4, even though 1.99999 was computed. As with the previous procedure, however, you would make matters worse more often than you would fix them. Let's suppose that, as a result of some previous calculations, you need the square root of 3.99996. You don't see this figure since each calculation falls



ot of 4 is not 2

Philip Stein

directly on the heels of the next. The correct answer is 1.99999, but the square root program gives you 2.00000. Just as large, just as wrong.

A base violation

The third source of error is conversion. Almost all minicomputers operate in binary internally. Although it is possible to exactly convert integers from decimal to binary and back, the same is not true of fractions or of numbers with both integer and fractional portions. In addition, frank errors in the conversion procedure due to coding bugs are always possible. Because of this, exact answers in binary may become inexact in decimal.

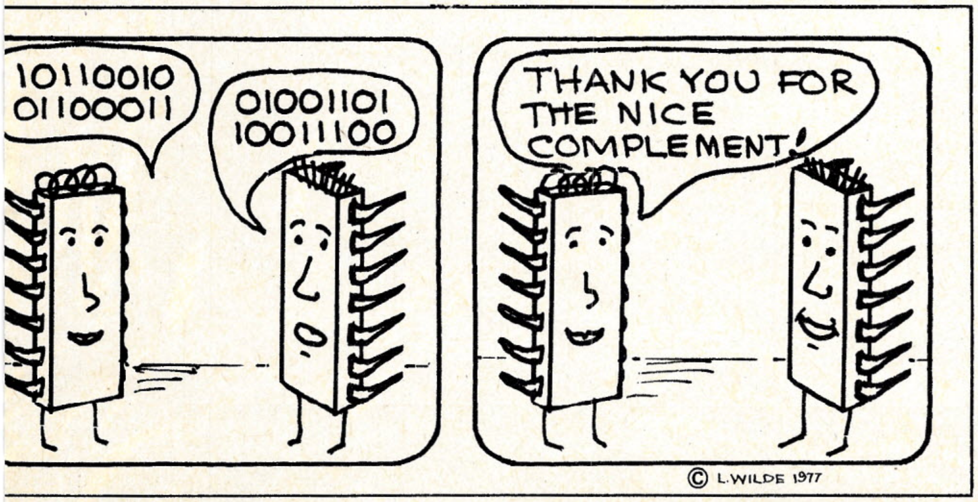
This source of error is avoidable to some extent. When you program, it pays to keep intermediate results in binary form. In *Fortran,* for example, use unformatted *reads* and *writes* when using mass storage to save temporary data which won't fit in memory. This leaves the data in binary and avoids two passes through the format converter. In *Basic,* binary I/O is usually not possible, but you may be abie to organize your programs so that they do not use more memory than is available. If your *Basic* supports program overlays but allows you to keep data, you may be able to compromise with many segments of code and only one data area.

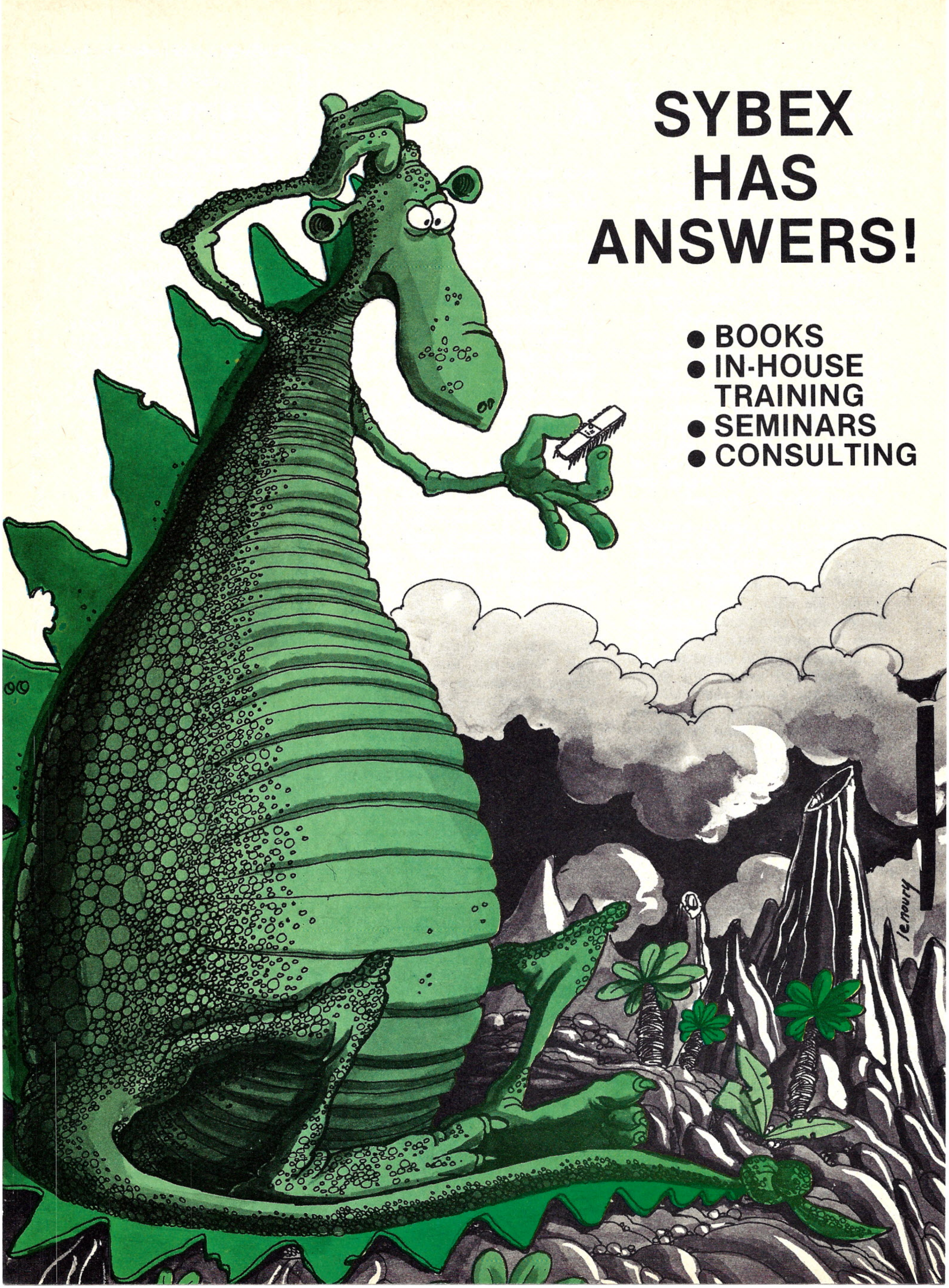
But what to do?

Basically, you just have to get used to the fact that computer errors and round-offs are part of life. As we just demonstrated, attempts to get rid of them almost always result in generating more. If you are teaching neophytes, maybe you could convert the outrageous proposition that the square root of 4 is not necessarily 2 into a strong lesson in numerical analysis and the limitations of computing everything digitally.

Another good exercise is to draw a graph showing the difference between the computer value of a function and the "correct" value (usually obtained from a reliable table). This chart should show some periodic structure in which the computed value oscillates around the correct value. You should be able to draw error bands showing the maximum error in the function routine. With this tool, you could demonstrate that rounding programs or approximations designed to give the best answers for certain inputs will almost certainly have larger error bands for all inputs. If they don't, you may have discovered something.

Reprinted with permission from *Computer Decisions*, Sep. 1975. Copyright 1975 by Hayden Publishing Co., Rochelle Park, N.J. ■




SYBEX
HAS
ANSWERS!
● BOOKS
● IN-HOUSE
TRAINING
● SEMINARS
● CONSULTING
lenoury

microprocessors: from chips to systems

by **Rodnay Zaks** · **$9.95**

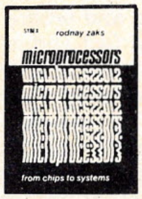
The basic book for understanding microprocessors and microcomputers. It presents all the fundamental concepts as well as the detailed application techniques. It will show you, in detail, how a microprocessor operates, critically review existing MPU's, explain how to interconnect the various chips into a system, and which alternatives are available. It is based on the author's experience in teaching microprocessors to more than 2000 persons, as well as developing systems. It will take you step by step until the actual interfacing to peripherals. 400p.

CONTENTS:

• Fundamental Concepts • Internal Operation of an MPU • System Chips • Comparative Microprocessor Evaluation • Assembling a System • Applications • Interfacing • Programming • System Development • The Future

ALSO AVAILABLE:

- CASSETTES (2 cassettes = 2½ hour + special book)
  - **S1** · INTRODUCTION TO MICROPROCESSORS **$29.95**
  - **S2** · MICROPROCESSOR PROGRAMMING **$29.95**
- MICROPROCESSOR INTERFACING TECHNIQUES, by **Austin Lesea** and **Rodnay Zaks.** Ref C207. From keyboard to CRT and Floppy Disk, including Trouble-Shooting **$9.95**
- MD- 10-language International Microprocessor Dictionary **$1.95**



FOR INFORMATION ON
IN-HOUSE TRAINING:

SYBEX

USA: SYBEX Inc, Publications Dept, 2161 Shattuck Ave, Berkeley, Ca 94704. Tel:(415) 848-8233. Telex: 336311.
EUROPE: SYBEX-Europe, 313 rue Lecourbe, 75015 Paris, France Tel: (1)8282502, Telex: 200858

ORDER FORM

Name ____________________

Position ____________________

Company ____________________

Address ____________________

____

Amount Enclosed ____________________
(in California, add tax)

☐BankAmericard ☐Mastercharge

Number ____________ Expires ________

Signature ____________________

Fast Shipping: add $1.00 (UPS) C201

Creative Computing

MICROPROCESSOR INTERFACING TECHNIQUES

Austin Lesea and **Rodnay Zaks.** Ref C207 - Available : Oct. 15 - **$9.95**

FROM KEYBOARD TO FLOPPY DISK:

The book takes you through the complete assembly of a microprocessor system: assembling a CPU; Input-output techniques; interfacing to a keyboard, LED, teletype, printer, floppy disk, CRT, cassette-tape; Industrial interfacing: analog-digital techniques; Communications; Busing and standards, including S100, IEEE488, CAMAC. A complete case-study is presented for a multi-channel communication system and a chapter is dedicated to trouble-shooting techniques. Actual interconnects are presented for a number of microprocessors, and, in particular the 8080 and the 6800.

ALSO AVAILABLE:

- CASSETTES (2 cassettes = 2½ hour + special book)
  - **S1** · INTRODUCTION TO MICROPROCESSORS **$29.95**
  - **S2** · MICROPROCESSOR PROGRAMMING **$29.95**
- MICROPROCESSORS: from chips to systems, by Rodnay Zaks
  The Basic microprocessor book. Ref C201 .... **$9.95**
- MD - 10-Language International Microprocrressor Dictionary **$1.95**
- SEMINAR BOOKS: each **$39.95**

B1-MICROPROCESSORS	B5- BIT-SLICES
B2-MICROPROCESSOR PROGRAMMING	B6- INDUSTRIAL TECHNIQUES
B3-MILITARY TECHNIQUES	B7- INTERFACING TECHNIQUES

SYBEX

FOR INFORMATION ON
IN-HOUSE TRAINING:

USA: SYBEX inc, 2161 Shattuck Ave, Berkeley, Ca 94704. Tel:(415) 848-8233. Telex: 336311
EUROPE: SYBEX-Europe, 313 rue Lecourbe, 75015 Paris, France Tel:(1)8282502, Telex: 200858

ORDER FORM

Name ____________________

Position ____________________

Company ____________________

Address ____________________

____

Amount Enclosed ____________________
(in California, add tax)

☐BankAmericard ☐Mastercharge

Number ____________ Expires ________

Signature ____________________

Fast Shipping: add $1.00 (UPS) C207

Creative Computing

THE RHETORIC OF THE COMPUTER

by
Barbara Marsh

When I was working in industry as a programmer-analyst some years ago, it seemed very clear to me that the public should be disabused of its ignorance about computers. I lived in the United States then, and public concern with the effects of computers on society was often aired. There were Senate hearings on the threat posed to privacy (which Americans hold to be a basic right) by data banks. Newspaper cartoons frequently expressed some aspects of the unease people felt when confronted with a decision said to have been made by a computer.

"Demystification" was my catchcry. No wonder people were scared of computers if they didn't understand how they worked. No wonder they were hostile when they knew that computers were used extensively in a war growing more unpopular each day. The magic phrase "Do not fold, spindle or multilate" appeared on the punched card which was the monthly telephone bill. In the days of the Vietnam war, protestors had seized upon the irony of corporate concern for the welfare of pieces of cardboard, and sometimes sported placards at anti-war demonstrations, saying "People: do not fold, spindle or mutilate."

I'd thought about my own socialization as a computer person. Nothing had been said, in formal or on-the-job training, about the bad public image of the computer, yet somehow we computer people felt that our specialized knowledge made us immune to the fears that ignorance bred. When I became involved in education, I carried with me the zeal for demystification. People must learn how computers worked, though by now I knew that we must also acknowledge that computers had been involved in some rather nasty applications, which people were quite entitled to fear.

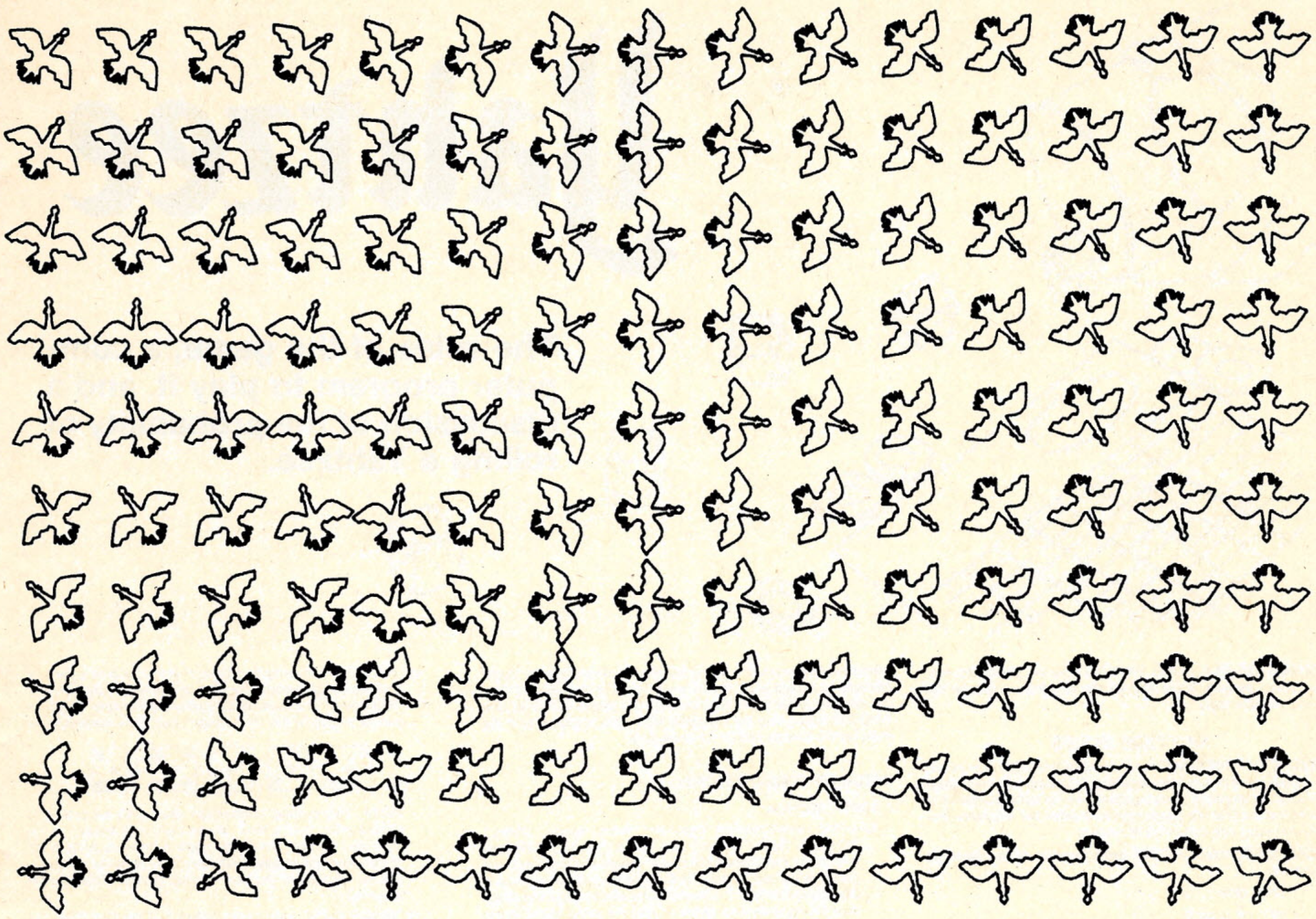
When the question of education about computers was raised, then, I first advocated computer awareness and a kind of computer literacy - courses for students, and for teachers, in what computers did and how they worked, aimed at demystifying the computer. I launched such a course for teachers, and hoped they would then be able to go out and demstify computers for their students. The level of understanding I thought appropriate was one of recognizing that most common applications of computers entailed isolating some real-world task, understanding the entities and processes involved, defining the processes as algorithms and then writing and testing a program to perform a representation of the processes on some numbers or other symbols that represent the entities. Most students I came across didn't express the need for a more physical understanding of how things were represented within the machine, so I never spent much time on hardware, or binary arithmetic or Boolean logic, for example. Teaching a little of high level language such as BASIC seemed to be a good way to get some of the processes across, andenabled students to write simple programs so that they could experience the various phases in setting and solving a problem using a computer as a tool.

*Center for the Study of Innovation in Education, LaTrobe University, Bundoora, Victoria, Australia 3083.

But by computer literacy I meant more, too, especially in Australia, where most of the teachers who were my students had never seen a punched card and where the visibility and impact of computers, both in fact and in the public imagination, is much lower than in the U.S.A. There really was an information gap to fill, for teachers ought to know the kinds of things computers were actually doing in classrooms and in various strands within educational research. So the course I put together covered their own discipline, education. To help in interpreting and doing empirical educational research, I tried to teach how to use a large statistical package intelligently, always realizing that without some background in statistics and some meaningful research problem in mind, this would have been a futile exercise. I taught some BASIC programming, where again it is clear that unless my students had at least one problem or class problem in mind, this tool also would be useless. The teachers had a chance to use some fairly low level computer assisted instruction, and we talked about some of the questions raised or answered by C.A.I., computer-managed instruction and such computer-aided augmentation of curriculum as using computer simulations or games.

I have no wish to disown this course, nor is there much in this range of topics I'd wish to change, but I've recently had some new insights about what teachers can offer their students. Most schoolteachers, especially those who graduated more than five years ago, will not have had any direct contact with computers in their undergraduate education. Even today, the social and physical sciences and mathematics are the only disciplines of which one can safely say that most undergraduates do some work on a computer — and then the kinds of work which are done are fairly limited.

By contrast, consider their students today and in the future. All of them watch a lot of television, from "Star Trek" to "The Bionic Woman". Computers have always been part of their world, and they are considerably less surprised at what computers can do than are teachers from an earlier generation. Also, it seems likely that in a few years we'll have a computer utility, along the lines of the telephone or even the electric power network, and also will find microcomputers built into many home appliances in a completely taken-for-granted way. Recreational uses such as electronic games are becoming ubiquitious, and the prices on units which attach to home television sets for computerized game-playing are dropping rapidly, as



GEESE

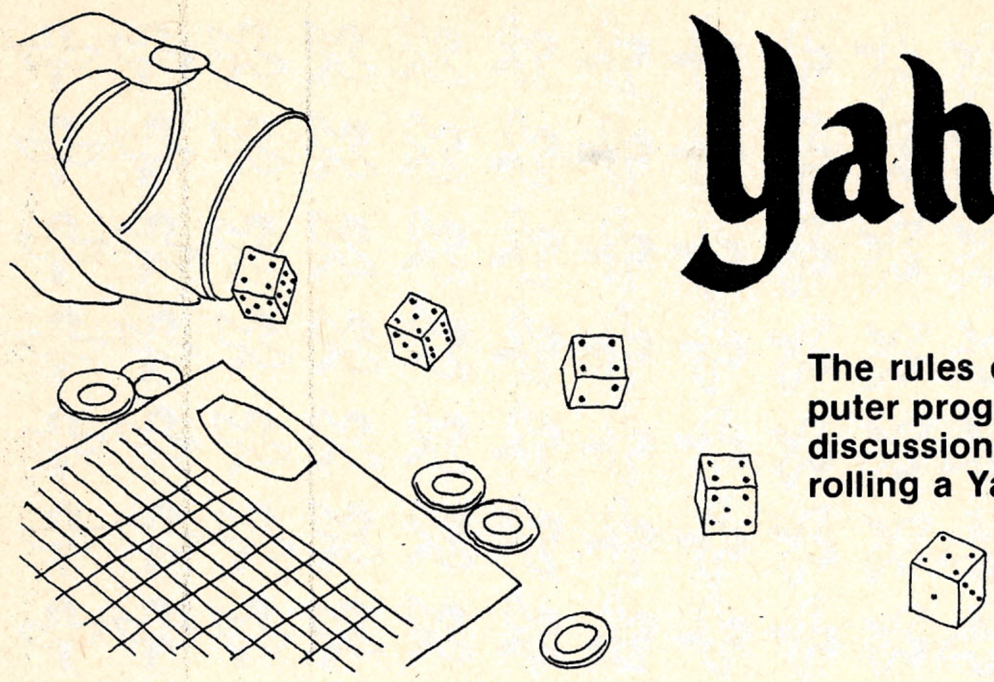
Bill Kolomyjec

calculator prices have already. So teachers may not need to teach their students how to use computers, nor to tell them what computers can do — kids may arrive in school with this knowledge, much as they arrive knowing how to use the telephone or how to work a T.V. set. Demystification might be a beginning for teachers, but their students can hardly be said to need it!

What then can teachers offer their students? It is my belief that as adults with a good deal more experience in the world, they may have perspectives which can be quite valuable. Perhaps I could best characterize what they might offer as an understanding of the "rhetoric" of the computer, the ways computers, seen as media, are able to persuade us. An analogy may be useful here. While television and newspapers can present the news or other information with great immediacy, insight and depth, these media can also slant information by a variety of means, from selection of which items to report and which to omit, through biased reporting, to rather more subtle means such as editing and the use of montage. Analogously, the computer mediates information for us. Certain kinds of information are not included in computer analyses, and others may be disloged by the means of data collection employed prior to computer processing. Some constraints are inherent in the knowledge representation systems used. Certain kinds of information can be handled superbly well by computers, and others may be distorted beyond recognition.

I believe people ought not emerge from schools at the mercy of what they see on television, what they read in the newspapers (if, in fact, they read), or what a computer analysis tells them is the case. The television may define what are "the issues", and computer analyses may provide "the answers," but we must try to provide an education which leaves room for students to make their own evaluations and decisions. They should be able to assess the appropriateness of the computer mediation of the information dealt with, and have some idea of where to look to find what computers leave out. I think one way to make this more likely to happen is to have teachers who are able to think about computers as persuasive media whose output must be evaluated. Most of the characteristics of computerized information systems are present in computer uses in education, such as empirical research data processing, the various kinds of computer assisted learning, and artificial intelligence, so a consideration of such applications could be used as one data base for a study of the rhetoric of the computer. I intend to do just that in my new course for teachers, and I hope that they'll find ways to communicate this approach to their students, whether in social studies, English or even mathematics classes. It's time to stop imagining that computer literacy begins and ends with programming and that computer awareness means discussing data banks as a threat to privacy. ■

Yahtzee



The rules of the game, a computer program to play it, and a discussion of the probability of rolling a Yahtzee.

NUMBER OF PLAYERS: YAHTZEE may be played by two, three or four persons. It can also be played solitaire trying for the highest possible score.

YAHTZEE RULES

OBJECT OF THE GAME: The object of YAHTZEE is to obtain the highest score for one or more games. The player with the highest total score for all games (up to 6 games) wins. An optional method of scoring can be that the player who wins earns the difference between his score and that of his opponents.

EQUIPMENT: Your YAHTZEE game consists of the following equipment:

1 DICE CUP
1 SET OF 5 DICE
1 YAHTZEE SCORE PAD
YAHTZEE BONUS CHIPS

Game Summary: In each turn a player may roll the dice up to 3 times in order to obtain a scoring combination. After rolling the dice, he **MUST** place a score or a zero in one of the boxes in the vertical column under the game he is playing. The game ends after 13 rounds, when all scoring boxes have been filled. The scores are totaled and combined with bonus points on the reverse side of the score pad to give a player his final score.

Before playing the game each player is given a score card and either a pencil or pen (not supplied) to keep his own score.

TO PLAY:

1. To determine who goes first, each player places all five dice in the dice cup and rolls out all the dice. The player rolling the highest total starts the game. Play then continues clockwise.
2. In each turn a player is allowed a **maximum** of 3 rolls of the dice, although he may stop after the first or second roll.
   A. For the first roll he must roll all 5 dice. The five dice are placed in the cup, the cup is shaken and the dice rolled out.
   B. For the second and third rolls the player may pick up ANY or ALL the dice and roll again. He peed not declare what he is trying to make (what box he is trying to score in) and may change his mind at any time.
   C. The dice are final after the third roll and MUST BE SCORED.

SCORING:

Each player has his own score card. He must make his own decision of what and where to score based on his individual strategy. In each game there are 13 scoring rounds. A player **MUST** place a score or a zero after each turn in one of the 13 scoring boxes in the vertical column under the game being played.

There are 13 scoring boxes on the score card. They are: "Aces," "Twos," "Threes," "Fours," "Fives," and "Sixes" in the Upper Section, and "3 of a Kind," "4 of a Kind," "Full House," "Small Straight," "Large Straight," "YAHTZEE" (five of a kind), and "Chance" in the Lower Section.

At the end of each turn the player must place a score or a zero in one of the scoring boxes. A scoring box can be used only once in each game. A player can place a zero at the end of a round rather than a score, if he believes it to be to his advantage. The boxes may be filled in any order, according to the player's best judgment.

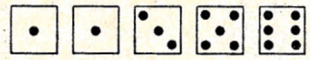
On the score card there is an "Upper Section" and a "Lower Section" which, when added together, will give you a grand total which is recorded on reverse side with bonus chips (if any) for your final score. There are possibilities to earn bonus points which count toward the final score. If a player attains a score of 63 in the Upper Section, he should enter the 35-point bonus in the "Bonus" box and add this to his Upper Section total. If a player is entitled to a YAHTZEE bonus chip (as explained later), this is totaled with the score on the reverse side of the score card.

SCORING COMBINATIONS:

1. **UPPER SECTION:**

a) In the Upper Section, there are "Aces" (ones), "Twos," "Threes," "Fours," "Fives," and "Sixes." If a player chooses to score in the Upper Section, he counts and adds only the dice with the same number and enters the total of the dice in the appropriate box. If a player at the end of a turn has these dice on the table and elects to take his score in the Upper Section, he would enter 9 in the "Threes" box.

b) The player may enter the total of ANY NUMBER of the same value dice in the appropriate box in the Upper Section. For example, a player's final dice are:



he may choose to score 2 in the "Aces" box, or 6 in the "Sixes".

c) If a player is unable to place a score in either the Upper or Lower Sections, he **MUST** enter a zero in the scoring box of his choice in either section.

d) If a player scores 63 points or more in the Upper Section, he earns a BONUS of 35 points and enters this in the "Bonus" box in the Upper Section. (For quick calculation, 63 can be obtained by scoring 3 of each number "Aces through sixes.")

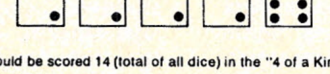
2. **LOWER SECTION:**

The Lower Section of the score card is played exactly as indicated.

a) "3 of a Kind" box may be filled in only if the dice show at least three of the same number. For example:

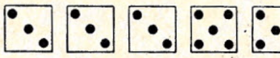
would be scored 18 (total of all dice) in the "3 of a Kind" box.

b) "4 of a Kind" box scores the total of all dice provided they include four of the same number. For example:



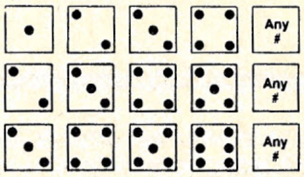
would be scored 14 (total of all dice) in the "4 of a Kind" box. NOTE: This could also be scored in the "3 of a Kind" box as 14 as it also meets the requirements for "3 of a Kind." The player must decide which box to score in as only one box can be scored in at the completion of a turn.

c) "Full House" box may be scored in when the dice show any combination of three of one number and two of another. Any "Full House" is scored as 25 points. For example, 25 points can be scored in the "Full House" box if the markers show

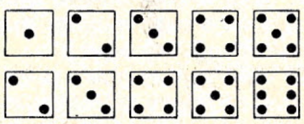


(NOTE: This scoring combination could also be scored in either (1) the "3 of a Kind" box in the Lower Section as 19, (2) in the "Threes" box in the Upper Section as 9, (3) in the "Fives" box in the Upper Section as 10, or (4) in the "Chance" box as 19. (Again, the player must decide which box to score in.)

d) "Small Straight" box can be scored in if the dice show any sequence of four numbers. Any "Small Straight" scores 30 points. The following are examples of "Small Straights":



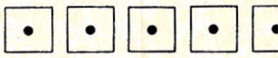
e) "Large Straight" box can be scored in if the dice show any sequence of five numbers. Any "Large Straight" scores 40 points. The following are examples of "Large Straights":



(NOTE: "Large Straights" may also be scored as 30 in the "Small Straight" box.)

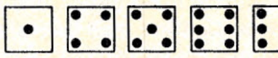
f) "YAHTZEE" box can be scored if the dice show five of the same number ("5 of a Kind"). A "YAHTZEE" scores 50 points.

One example of a "YAHTZEE" is:



Refer to "YAHTZEE Bonus" and "YAHTZEE used as a Joker" to score more than one "YAHTZEE" in a game.

g) "Chance" box scores the total of all markers and is the one chance to score any combination. For example:



could be scored as 22 (the total of all markers) in the "Chance" box if the player chooses to score in "Chance".

h) YAHTZEE Bonus Chips: A player receives a Bonus chip (valued at 100 points) for his second and subsequent YAHTZEE if, and only if, the first YAHTZEE was scored as 50 in the

or 40 points in the "Large Straight" box. If all the boxes in the Lower Section are filled, he must enter a zero in a box of his choice in the Upper Section. HE MAY ALSO COLLECT A BONUS CHIP IF HE IS ENTITLED TO IT.

If the YAHTZEE is scored in the Upper Section it scores the **total** of the dice. (Five fours scores 20 in the "Fours" box.) The player, in addition, would collect a bonus chip if he is entitled to it.

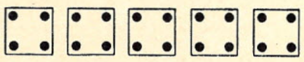
"YAHTZEE" box. (If he had previously scored a zero in the "YAHTZEE" box, he is NOT entitled to any bonus chips in that game.)At the end of the game, the chips are totaled as 100 points each and are added to the score on the reverse side of the score card.

"YAHTZEE used as a Joker":

A YAHTZEE may be used as a joker in the Lower Section when both of the following conditions exist:

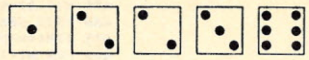
1) The YAHTZEE box has been previously filled with 50 or zero.
2) The appropriate box in the Upper Section has been filled. For example, if the YAHTZEE consists of five 4's and the "Fours" box has previously been filled.

The player marks his score in any of the boxes in the Lower Section as follows: If the markers were



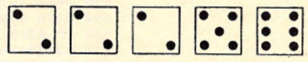
he may score the total of all five markers, which in this case equals 20 points, in any one of the following boxes: "3 of a Kind," "4 of a Kind," or "Chance." Or, he may score 25 points in the "Full House" box, 30 points in the "Small Straight" box

EXAMPLE OF PLAY: If on the first roll of the dice, a player has



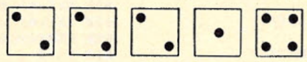
he might choose to go for "Twos" in the Upper Section of the score card or "3 of a Kind" (in this case, 2's) in the Lower Section of the score card. The player would then leave the dice marked "2" on the table, pick up the 1, 3 and 6 dice and attempt to toss more 2's on his second roll.

If on the second roll of the dice, he has



he might stop there and enter 6 in the "Twos" box in the Upper Section of the score card or 17 (total of all five dice) in the "3 of a Kind" box in the Lower Section. Or he might choose to roll again in the hope of getting one or even two more 2's.

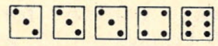
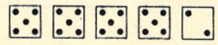
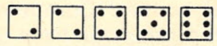
If the player chooses to roll again, he picks up the 5 and 6 dice only and tosses for his third and last roll. If, on his third roll, he has



he could enter his score as 6 in the "Twos" box, 1 in the "Aces" box or 4 in the "Fours" box in the Upper Section or 11 (total of all five dice) in the "3 of a Kind" box or "Chance" box in the Lower Section.

STRATEGY:

If you are not as lucky as your opponent, try to beat him with strategy. Here are a few examples:

If at the completion of each turn, your dice show:	*You have the choice of scoring the following boxes, if open:*
	9 points in "Threes" box in Upper Section; or 19 points (total of all dice) in "3 of a Kind" or "Chance" box in Lower Section.
	20 points in "Fives" box in Upper Section; or 22 points (total of all dice) in "3 of a Kind" or "4 of a Kind" box, or "Chance" box in Lower Section.
	19 points (total of all dice) in "Chance" in Lower Section, or 4 points in "Twos" box in Upper Section. This last choice, however, puts you 2 points below (63) for Upper Section bonus of 35 points. To overcome this deficit, you would have to score either 4 "Threes," "Fours," "Fives" or "Sixes" during the game to earn the bonus.

If, in any of the above examples, all appropriate boxes were already filled, you would have to score a zero in a blank box. It would be best if you could take a zero in an open box of the Upper Section without ruining your chances for making 63 to earn the 35 point bonus. If this cannot be done, a player would have to use his judgment in placing the zero so as to lose the minimum number of points.

Yahtzee

PLAYERS NAME____________________

SCORE CARD

MINIMUM REQUIRED FOR BONUS	HOW TO SCORE	GAME #1	GAME #2	GAME #3	GAME #4	GAME #5	GAME #6
Aces = 3	COUNT AND ADD ONLY ACES						
Twos = 6	COUNT AND ADD ONLY TWOS						
Threes = 9	COUNT AND ADD ONLY THREES						
Fours = 12	COUNT AND ADD ONLY FOURS						
Fives = 15	COUNT AND ADD ONLY FIVES						
Sixes = 18	COUNT AND ADD ONLY SIXES						
TOTAL 63	➽						
Bonus IF 63 OR OVER	SCORE 35						
TOTAL OF UPPER HALF	➽						

3 of a kind	ADD TOTAL OF ALL DICE						
4 of a kind	ADD TOTAL OF ALL DICE						
Full House	SCORE 25						
Sm. Straight (Sequence) of 4	SCORE 30						
Lg. Straight (Sequence) of 5	SCORE 40						
YAHTZEE 5 of a kind	SCORE 50						
Chance	SCORE TOTAL OF ALL 5 DICE						
TOTAL OF LOWER HALF	➽						
TOTAL OF UPPER HALF	➽						
GRAND TOTAL	➽						

6J74

Yahtzee continued....

This is the BASIC version of the dice game Yahtzee. The game is played according to the standard rules. The program was written for the Timeshared-8 Edusystem 50, with 4K of memory for each user, by Steve Elias, 25 Turning Mill Road, Lexington, MA 02173.

Following the game is a fascinating, in-depth article on "The Probability of a Yahtzee: Analysis and Computation," reprinted with permission from *School Science and Mathematics.*

Program Listing

```
5 'PROGRAM WRITTEN BY STEVE ELIAS.
7 'LEXINGTON HIGH SCHOOL, LEXINGTON, MA
10 RANDOM
15 DIM C(5),K(14,7),G(5),F(4),H2(6),L2(6),B5(7),B8(7),A$(7)
20 PRINT CHR$(7)\PRINT CHR$(7)\PRINT CHR$(7)
25 PRINT"THIS IS THE GAME OF YAHTZEE, DO YOU WANT COMMENTS";
30 INPUT B$\IF B$="NO" THEN 45\IF B$="YES" THEN 40
35 GOSUB 935\GOTO 25
40 GOSUB 995
45 PRINT"HOW MANY PLAYERS";\INPUT N
50 IF N<1 THEN 45\IF N<=7 THEN 75
55 PRINT"THE MAXIMUM NUMBER OF PLAYERS IS 7."
70 GOTO 45
75 FOR A0=1 TO N
80 PRINT"NAME OF PLAYER #";A0;\INPUT A$(A0)\NEXT A0
85 PRINT"WOULD YOU LIKE A SCORECARD";\INPUT B$
90 IF B$="NO" THEN 100\IF B$="YES" THEN 945
95 GOSUB 935\GOTO 85
100 H=0\PRINT\A=A+1\IF A<>N+1 THEN 105\A=1
105 FOR B=1 TO 5\C(B)=INT(6*RND(0))+1\NEXT B\PRINT
110 PRINT\PRINT A$(A);"'S NUMBERS:  ";\H=H+1
115 PRINT C(1);C(2);C(3);C(4);C(5)\IF H=3 THEN 210
120 PRINT"HOW MANY";\INPUT E
125 IF E<>25 THEN 130\GOSUB 1020\GOTO 120
130 IF E>5 THEN 120
135 IF E<0 THEN 120\IF INT(E)<>E THEN 120
140 IF E=0 THEN 210\IF E=5 THEN 105
145 FOR F3=1 TO 4\F(F3)=0\NEXT F3
150 PRINT"WHICH DICE";
155 ON E GOTO 160,165,170,175
160 INPUT F(1)\GOTO 180
165 INPUT F(1),F(2)\GOTO 180
170 INPUT F(1),F(2),F(3)\GOTO 180
175 INPUT F(1),F(2),F(3),F(4)
180 FOR F1=1 TO 4\F(F1)=INT(F(F1))\IF F(F1)>5 THEN 150
185 IF F(F1)<0 THEN 150\NEXT F1
190 FOR F2=1 TO 4\IF F(F2)=0 THEN 200
195 C(F(F2))=INT(6*RND(0))+1
200 NEXT F2
205 GOTO 110
210 PRINT A$(A);"--SCOREBOARD NUMBER";
215 INPUT I
220 IF I<>25 THEN 225\GOSUB 1020\GOTO 210
225 IF I<1 THEN 210\I=INT(I)\IF I<14 THEN 245
230 PRINT"CONFIRM";\INPUT B$
235 IF B$="NO" THEN 210\IF B$="YES" THEN 735
240 GOSUB 935\GOTO 230
245 IF K(I,A)=0 THEN 260
250 PRINT A$(A);", YOU HAVE ALREADY USED ROW #";I;
255 PRINT"YOU HAVE"K(I,A);"POINTS IN ROW #";I\GOTO 210
260 IF I>6 THEN 305
265 REM***CATEGORIES ONE THROUGH SIX
270 K=0
275 FOR G=1 TO 5\IF C(G)<>I THEN 280\K=K+I
280 NEXT G\IF K=0 THEN 295\IF K<>1 THEN 300
285 PRINT A$(A);", YOU NOW HAVE 1 POINT IN ROW # 1 (ACES)"
290 K(1,A)=1 \ GOTO 720
295 K(I,A)=.4\GOTO 620
300 K(I,A)=K\GOTO 620
305 ON I-6 GOTO 315,320,400,485,485,325,605
310 REM***3 OF A KIND, 4 OF A KIND, YAHTZEE
315 R=3\T=C(1)+C(2)+C(3)+C(4)+C(5)\GOTO 330
320 R=4\T=C(1)+C(2)+C(3)+C(4)+C(5)\GOTO 330
325 R=5\T=50
330 FOR H3=1 TO 6\H2(H3)=0\NEXT H3
335 FOR H=1 TO 6\FOR H1=1 TO 5
340 IF C(H1)<>H THEN 350
345 H2(H)=H2(H)+1
350 NEXT H1\NEXT H
355 FOR H4=1 TO 6
360 IF H2(H4)>=R THEN 370
365 NEXT H4\K(I,A)=.4\GOTO 620
370 IF K(I,A)=50 THEN 380
375 K(I,A)=T\IF I<>12 THEN 395
380 PRINT CHR$(7)\PRINT CHR$(7)\FOR V4=1 TO 7\PRINT CHR$(7);\NEXT V4
385 PRINT"CONGRATULATIONS ";A$(A);" YOU HAVE A YAHTZEE!!!"
390 PRINT CHR$(7)\PRINT CHR$(7)\FOR V4=1 TO 7\PRINT CHR$(7);\NEXT V4
395 GOTO 620
400 REM***FULL HOUSE
405 L4=0\L5=0\FOR L=1 TO 6\L2(L)=0\NEXT L
410 FOR L=1 TO 6\FOR L1=1 TO 5
415 IF C(L1)<>L THEN 425
420 L2(L)=L2(L)+1
425 NEXT L1\NEXT L
430 FOR L3=1 TO 6
435 IF L5=1 THEN 450
440 IF L2(L3)<>2 THEN 450
445 L5=L5+1\GOTO 455
450 IF L2(L3)<>3 THEN 460
455 L4=L4+1
460 NEXT L3
465 IF L4=2 THEN 475
470 K(9,A)=.4\GOTO 620
475 K(9,A)=25\GOTO 620
480 GOTO 620
485 REM***PUT NUMBERS IN ORDER, STRAIGHTS
490 FOR M1=1 TO 5\FOR M2=1 TO M1
495 IF C(M1)>=C(M2) THEN 505
500 Z=C(M1)\C(M1)=C(M2)\C(M2)=Z
505 NEXT M2\NEXT M1
510 G=0\M3=0\G3=0
515 G2=0\IF I=11 THEN 545
520 G=G+1\IF G=15 THEN 600
525 READ G(1),G(2),G(3),G(4),G(5)\GOTO 570
530 DATA 1,2,3,4,7,2,3,4,5,7,7,3,4,5,6,7,2,3,4,5,7,1,2,3,4
535 DATA 1,7,2,3,4,1,2,7,3,4,1,2,3,7,4,2,7,3,4,5,2,3,4,7,5
540 DATA 2,3,7,4,5,3,7,4,5,6,3,4,7,5,6,3,4,5,7,6
545 IF G3>1 THEN 600
550 G3=G3+1\G5=0
555 FOR G4=G3 TO (G3+4)
560 G5=G5+1\G(G5)=G4
565 NEXT G4
570 FOR G1=1 TO 5
575 IF C(G1)<>G(G1) THEN 580\G2=G2+1
580 NEXT G1
585 IF G2=I-6 THEN 590\GOTO 515
590 IF I=11 THEN 595\K(I,A)=30\GOTO 620
595 K(I,A)=40\GOTO 620
600 K(I,A)=.4\GOTO 620
605 REM***CHANCE
610 Y1=0
615 FOR Y=1 TO 5\Y1=C(Y)+Y1\NEXT Y\K(13,A)=Y1
620 PRINT A$(A);", YOU NOW HAVE";INT(K(I,A));"POINTS IN ROW #";I;", ";
625 PRINT"(";
630 IF I>6 THEN 640
635 ON I GOTO 645,650,655,660,665,670
640 ON I-6 GOTO 675,680,685,690,695,700,705
645 PRINT"ACES";\GOTO 710
650 PRINT"TWOS";\GOTO 710
655 PRINT"THREES";\GOTO 710
660 PRINT"FOURS";\GOTO 710
665 PRINT"FIVES";\GOTO 710
670 PRINT"SIXES";\GOTO 710
675 PRINT"3 OF A KIND";\GOTO 710
680 PRINT"4 OF A KIND";\GOTO 710
685 PRINT"FULL HOUSE";\GOTO 710
690 PRINT"SM. STRAIGHT";\RESTORE*\GOTO 710
695 PRINT"LARGE STRAIGHT";\GOTO 710
700 PRINT"YAHTZEE";\GOTO 710
705 PRINT"CHANCE";
710 IF E=25 THEN 1030\IF I0=25 THEN 1030
715 PRINT")"
720 FOR B=1 TO N\FOR B1=1 TO 13
725 IF K(B1,B)=0 THEN 100
730 NEXT B1\NEXT B
735 FOR B3=1 TO N\B5(B3)=0\B8(B3)=0\NEXT B3
740 PRINT
745 FOR B3=1 TO  N\FOR B4=1 TO 6
750 B5(B3)=B5(B3)+INT(K(B4,B3))
755 NEXT B4\NEXT B3
760 FOR B6=1 TO N\FOR B7=7 TO 13
765 B8(B6)=B8(B6)+INT(K(B7,B6))
770 NEXT B7\NEXT B6
775 FOR B3=1 TO N
780 IF B5(B3)<63 THEN 790
785 K(14,B3)=35
790 NEXT B3
795 GOSUB 940
800 FOR A9=1 TO N
805 PRINT\PRINT\PRINT
810 PRINT TAB(19);A$(A9);"'S TOTALS"\PRINT TAB(19);"****************"
815 PRINT
820 PRINT"ACES-----------";INT(K(1,A9))
825 PRINT"TWOS-----------";INT(K(2,A9))
830 PRINT"THREES---------";INT(K(3,A9))
835 PRINT"FOURS----------";INT(K(4,A9))
840 PRINT"FIVES----------";INT(K(5,A9))
845 PRINT"SIXES----------";INT(K(6,A9))\PRINT
850 PRINT"3 OF A KIND----";INT(K(7,A9))
855 PRINT"4 OF A KIND----";INT(K(8,A9))
860 PRINT"FULL HOUSE-----";INT(K(9,A9))
865 PRINT"SM. STRAIGHT---";INT(K(10,A9))
870 PRINT"LG. STRAIGHT---";INT(K(11,A9))
875 PRINT"YAHTZEE--------";INT(K(12,A9))
880 PRINT"CHANCE---------";INT(K(13,A9))
885 PRINT\PRINT"TOTAL OF UPPER HALF------";B5(A9)
890 PRINT\PRINT"BONUS--------------------";K(14,A9)
895 PRINT\PRINT"TOTAL OF LOWER HALF------";B8(A9)
900 PRINT
905 B9=B5(A9)+B8(A9)+K(14,A9)
910 PRINT A$(A9);"'S GRAND TOTAL---------------------------";B9
915 GOSUB 940
920 NEXT A9
925 IF I>13 THEN 210
930 STOP
935 PRINT"ANSWER WITH YES OR NO"\RETURN
940 FOR E=1 TO 72\PRINT"-";\NEXT E\PRINT\RETURN
945 GOSUB 940
950 PRINT"1**ACES"\GOSUB 940\PRINT"2**TWOS"\GOSUB 940
955 PRINT"3**THREES"\GOSUB 940\PRINT"4**FOURS"\GOSUB 940
960 PRINT"5**FIVES"\GOSUB 940\PRINT"6**SIXES"\GOSUB 940
965 PRINT"7**3 OF A KIND"\GOSUB 940\PRINT"8**4 OF A KIND"\GOSUB 940
970 PRINT"9**FULL HOUSE"\GOSUB 940\PRINT"10**SM. STRAIGHT"\GOSUB 940
975 PRINT"11**LG. STRAIGHT"\GOSUB 940\PRINT"12**YAHTZEE"\GOSUB 940
980 PRINT"13**CHANCE"\GOSUB 940
985 FOR E1=1 TO 8\PRINT\NEXT E1\PRINT"TEAR OFF SCORECARD"
990 SLEEP 18\GOTO 100
995 PRINT"TOTALS WILL BE PRINTED IF YOU RESPOND TO"
1000 PRINT"'SCOREBOARD NUMBER?' WITH A NUMBER GREATER THAN 13."
1005 PRINT"TO SEE WHAT SCOREBOARD OPTIONS YOU HAVE NOT USED,"
1010 PRINT"RESPOND TO 'HOW MANY' OR 'SCOREBOARD NUMBER' WITH 25."
1015 RETURN
1020 PRINT A$(A);", YOU HAVE THE FOLLOWING SCOREBOARD OPTIONS LEFT:"
1025 I0=25\FOR I=1 TO 13\IF K(I,A)=0 THEN 630\GOTO 1035
1030 PRINT"  ";
1035 NEXT I\PRINT"."\I0=0\RETURN
1040 END
```

Sample Run

—Game opening—

```
THIS IS THE GAME OF YAHTZEE, DO YOU WANT COMMENTS? YES
TOTALS WILL BE PRINTED IF YOU RESPOND TO
'SCOREBOARD NUMBER?' WITH A NUMBER GREATER THAN 13.
TO SEE WHAT SCOREBOARD OPTIONS YOU HAVE NOT USED,
RESPOND TO 'HOW MANY' OR 'SCOREBOARD NUMBER' WITH 25.
HOW MANY PLAYERS? 2
NAME OF PLAYER # 1 ? STEVE
NAME OF PLAYER # 2 ? SUSAN
WOULD YOU LIKE A SCORECARD? NO


STEVE'S NUMBERS:   4  2  3  3  3
HOW MANY? 2
WHICH DICE? 1,2

STEVE'S NUMBERS:   4  1  3  3  3
HOW MANY? 2
WHICH DICE? 1,2

STEVE'S NUMBERS:   1  4  3  3  3
STEVE--SCOREBOARD NUMBER? 3
STEVE, YOU NOW HAVE 9 POINTS IN ROW # 3 , (THREES)


SUSAN'S NUMBERS:   1  3  5  2  3
HOW MANY? 3
WHICH DICE? 1,3,4

SUSAN'S NUMBERS:   6  3  6  2  3
HOW MANY? 1
WHICH DICE? 4

SUSAN'S NUMBERS:   6  3  6  6  3
SUSAN--SCOREBOARD NUMBER? 9
SUSAN, YOU NOW HAVE 25 POINTS IN ROW # 9 , (FULL HOUSE)


STEVE'S NUMBERS:   3  4  6  3  4
HOW MANY? 3
WHICH DICE? 1,3,4

STEVE'S NUMBERS:   5  4  4  1  4
HOW MANY? 2
WHICH DICE? 1,4

STEVE'S NUMBERS:   4  4  4  4  4
STEVE--SCOREBOARD NUMBER? 10-2

CONGRATULATIONS STEVE YOU HAVE A YAHTZEE!!!

STEVE, YOU NOW HAVE 50 POINTS IN ROW # 12 , (YAHTZEE)
```

—At the end of the game—

```
SUSAN'S NUMBERS:   5  3  4  5  6
SUSAN--SCOREBOARD NUMBER? 25
SUSAN, YOU HAVE THE FOLLOWING SCOREBOARD OPTIONS LEFT:
ACES  LARGE STRAIGHT  YAHTZEE  .
SUSAN--SCOREBOARD NUMBER? 12
SUSAN, YOU NOW HAVE 0 POINTS IN ROW # 12 , (YAHTZEE)


STEVE'S NUMBERS:   3  1  2  3  4
HOW MANY? 25
STEVE, YOU HAVE THE FOLLOWING SCOREBOARD OPTIONS LEFT:
TWOS  SIXES  .
HOW MANY? 4
WHICH DICE? 1,2,4,5

STEVE'S NUMBERS:   2  3  2  1  6
HOW MANY? 3
WHICH DICE? 2,4,5

STEVE'S NUMBERS:   2  1  2  2  2
STEVE--SCOREBOARD NUMBER? 2
STEVE, YOU NOW HAVE 8 POINTS IN ROW # 2 , (TWOS)


SUSAN'S NUMBERS:   3  5  4  5  6
HOW MANY? 1
WHICH DICE? 2

SUSAN'S NUMBERS:   3  2  4  5  6
HOW MANY? 0
SUSAN--SCOREBOARD NUMBER? 11
SUSAN, YOU NOW HAVE 40 POINTS IN ROW # 11 , (LARGE STRAIGHT)


STEVE'S NUMBERS:   1  5  2  5  1
HOW MANY? 5


STEVE'S NUMBERS:   5  1  2  5  1
HOW MANY? 5

STEVE'S NUMBERS:   1  3  3  4  4
STEVE--SCOREBOARD NUMBER? 6
STEVE, YOU NOW HAVE 0 POINTS IN ROW # 6 , (SIXES)


SUSAN'S NUMBERS:   6  2  3  4  4
HOW MANY? 5

SUSAN'S NUMBERS:   3  1  6  2  6
HOW MANY? 4
WHICH DICE? 1,3,4,5

SUSAN'S NUMBERS:   5  1  6  2  6
SUSAN--SCOREBOARD NUMBER? 1
SUSAN, YOU NOW HAVE 1 POINT IN ROW # 1 (ACES)
-----------------------------------------------------------------------

                STEVE'S TOTALS
                ***************

ACES----------- 2
TWOS----------- 8
THREES--------- 9
FOURS---------- 12
FIVES---------- 20
SIXES---------- 0

3 OF A KIND---- 19
4 OF A KIND---- 0
FULL HOUSE----- 25
SM. STRAIGHT--- 30
LG. STRAIGHT--- 40
YAHTZEE-------- 50
CHANCE--------- 10

TOTAL OF UPPER HALF------ 51

BONUS-------------------- 0

TOTAL OF LOWER HALF------ 174

STEVE'S GRAND TOTAL-------------------------- 225
-----------------------------------------------------------------------

                SUSAN'S TOTALS
                ***************

ACES----------- 1
TWOS----------- 4
THREES--------- 6
FOURS---------- 16
FIVES---------- 15
SIXES---------- 18

3 OF A KIND---- 29
4 OF A KIND---- 13
FULL HOUSE----- 25
SM. STRAIGHT--- 30
LG. STRAIGHT--- 40
YAHTZEE-------- 0
CHANCE--------- 20

TOTAL OF UPPER HALF------ 60

BONUS-------------------- 0

TOTAL OF LOWER HALF------ 157

SUSAN'S GRAND TOTAL-------------------------- 217
-----------------------------------------------------------------------
```

READER CHALLENGE

Readers: using your handy-dandy computer or calculator, can you determine the following:

1. The probability of rolling each type of scoring combination in the game of Yahtzee.
2. A set of "rules of thumb" for the best way to play the game, taking into account both probabilities, expected outcomes, and what has happened in preceding moves.
3. The expected score, assuming one followed the heuristics in (2).

The Probability of a Yahtzee: Analysis and Computation

David R. Duncan and Bonnie H. Litwiller

Associate Professors of Mathematics
University of Northern Iowa
Cedar Falls, Iowa 50613

Teachers of units concerning probability are interested in examples of probabilistic situations in a non-classroom setting. Games are a rich source of such probabilities. Many people enjoy playing a commercial mathematical game called Yahtzee. A Yahtzee player receives points for achieving various specified numerical combinations of dice during the three rolls which constitute his turn. The winner of the game is the player who accumulates the most points in 13 turns.

The purpose of this article is to examine the probabilities of one particular combination of the dice, that of having 5 of a kind (all 5 dice show the same number). This particular combination of dice is called a "Yahtzee" (the same name as the game). This combination is the most valuable one in terms of accumulating points during a game.

A turn for each player consists of the following:

1. He rolls all 5 dice at once.
2. If he is satisfied with the outcome, he may conclude his turn with the one roll. If he wishes to roll again, he may pick up *any* or *all* of the dice and re-roll them.
3. If he is satisfied with the new outcome, he may conclude his turn with his second roll. If he wishes to roll a third time, he may pick up *any* or *all* of the 5 dice and re-roll again. The third roll is the termination of his turn.

Suppose that a player decides that his goal on a particular turn is to roll a Yahtzee. The following six outcomes are possible on the first roll. (Here and henceforth, the Roman numerals will denote outcomes of the first roll of a turn.)

I. All 5 dice are alike. This yields a Yahtzee on the very first roll.
II. There are 4 dice alike and 1 die is different.
III. There are 3 dice alike and neither of the other two matches the 3 of a kind; however, they may match each other.
IV. Two dice show one numerical value, two other dice show a second numerical value, and the fifth die shows a third numerical value.
V. Two dice are alike; the other 3 are *all* different from each other and they also differ from the given pair.
VI. The 5 dice are all different.

Depending upon the situation achieved on the first roll, let us assume that the player will perform the second roll as follows: (Here and henceforth the capital letters will denote outcomes of the second roll of a turn.)

I. Since the player has achieved Yahtzee, he will not roll again.
II. He will roll the fifth die (the one die which was different). The possible outcomes are:
   A. The fifth die will match the other 4 thus producing a Yahtzee.
   B. The fifth die will not match.
III. The player will leave intact the 3 dice which are alike and roll the other two. The possible outcomes are:
   A. Both match the 3 of a kind thus yielding a Yahtzee.
   B. One die matches the 3 of a kind.
   C. No matches result with the original 3 of a kind.
IV. The player will leave one of the pairs, pick up the other three dice and roll them again. The possible outcomes are:
   A. All 3 dice match the original pair.
   B. There are 2 dice which match the original pair.
   C. There is one die which matches the original pair.
   D. The 3 dice all show the same number which is different from the original pair.
   E. The 3 dice do not *all 3* show the same number and none of them matches the original pair.
V. The player will leave the pair and re-roll the other 3 dice. The same 5 outcomes occur as in IV.
VI. The player will re-roll all 5 dice. For this case, outcomes A, B, C, D, E, and F are the same as I, II, III, IV, V, and VI.

On the third roll, the player will leave the maximal number of dice which are the same and roll the others. He either gets a Yahtzee or he does not. Either way his turn is concluded. Note in IV, D and V, D the player will leave the 3 of a kind and will now re-roll the original pair.

For the above given outcomes we shall now compute the appropriate probabilities using the fundamental principle of counting. In each case, the denominator of the probability is computed by observing that if k-distinct fair dice are rolled, 6^k equally likely outcomes are possible.

I. $\binom{6}{1} = 6$, the number of ways of choosing the number shown on all 5 dice.

$$P(\mathrm{I}) = \frac{6}{6^5} = \frac{1}{6^4}$$

II. $\binom{5}{4} = 5$, the number of ways of selecting the 4 matching dice.

$\binom{6}{1} = 6$, the number of ways of selecting the number to be shown on the four dice.

$\binom{5}{1} = 5$, the number of ways of selecting the number for the fifth die.

$$P(\mathrm{II}) = \frac{5 \cdot 6 \cdot 5}{6^5}$$

III. $\binom{5}{3} = 10$, the number of ways of selecting the three matching dice.

$\binom{6}{1} = 6$, the number of ways of selecting the number to be shown on these three matching dice.

$\binom{5}{1}^2 = 25$, the number of possible outcomes for the other two dice.

$$P(\mathrm{III}) = \frac{10 \cdot 6 \cdot 25}{6^5}$$

IV. $\binom{6}{2} = 15$, the number of ways of selecting the two numbers for the two pairs.

$\binom{5}{2} = 10$, the number of ways of selecting the two dice for the first pair.

$\binom{3}{2} = 3$, the number of ways of selecting the two dice for the second pair.

$\binom{4}{1} = 4$, the number of possible outcomes for the fifth die.

$$P(\mathrm{IV}) = \frac{15 \cdot 10 \cdot 3 \cdot 4}{6^5}$$

V. $\binom{5}{2} = 10$, the number of ways of selecting the two matching dice.

$\binom{6}{1} = 6$, the number of ways of selecting the number to be shown on these two dice.

$\binom{5}{1}\binom{4}{1}\binom{3}{1} = 60$, the number of possible outcomes for the remaining dice.

$$P(\mathrm{V}) = \frac{10 \cdot 6 \cdot 60}{6^5}$$

VI. 6 is the number of ways of choosing a number for the first die.
5 is the number of ways of choosing a number for the second die.
4 is the number of ways of choosing a number for the third die.
3 is the number of ways of choosing a number for the fourth die.
2 is the number of ways of choosing a number for the fifth die.

$$P(\mathrm{VI}) = \frac{6 \cdot 5 \cdot 4 \cdot 3 \cdot 2}{6^5}$$

We now investigate probabilities concerning the second roll:
Assuming that II has occurred:

A. $P(A) = 1/6$
B. $P(B) = 5/6$. In this case, the probability of a Yahtzee on the third roll is $1/6$, or $P(Y) = 1/6$, on roll 3.

Assuming III has occurred:

A. $P(A) = (1/6)^2$
B. $\binom{2}{1} = 2$, the number of ways of choosing the die to match the 3 of a kind.

$\binom{5}{1} = 5$, the number of ways of choosing the number for the non-matching die.

$P(B) = \dfrac{2 \cdot 5}{6^2}$. In this case $P(Y) = 1/6$, on roll 3.

C. $P(C) = (5/6)^2$. In this case $P(Y) = (1/6)^2$, on roll 3.

Assuming IV has occurred:

A. $P(A) = (1/6)^3$

B. $\binom{3}{2} = 3$, the number of ways of choosing the two matching dice.

$\binom{5}{1} = 5$, the number of ways of choosing the number for the other die.

$P(B) = \dfrac{3 \cdot 5}{6^3}$. In this case, $P(Y) = 1/6$ on roll 3.

C. $\binom{3}{1} = 3$, the number of ways of choosing the matching die.

$\binom{5}{1}^2 = 25$, the number of ways of choosing the numbers for the *other* two dice.

$P(C) = \dfrac{3 \cdot 25}{6^3}$. In this case, $P(Y) = (1/6)^2$ on roll 3.

D. $\binom{5}{1} = 5$, the number of ways of choosing the number for the three of a kind.

$P(D) = 5/6^3$. In this case, the $P(Y) = (1/6)^2$ on roll 3.

E. $\binom{5}{1}^3 - 5 = 120$, the number of ways of selecting 3 numbers, not all 3 alike, and none matching the original pair. Note that the (-5) term occurs to eliminate the case that the three dice all show the same number.

$P(E) = 120/6^3$. In this case, $P(Y) = (1/6)^3$ on roll 3.

Assuming that V has occurred, the probabilities of subsequent events are exactly the same as for IV.

Assuming that VI has occurred:

$P(A) = P(\mathrm{I}) = 6/6^5 = 1/6^4$

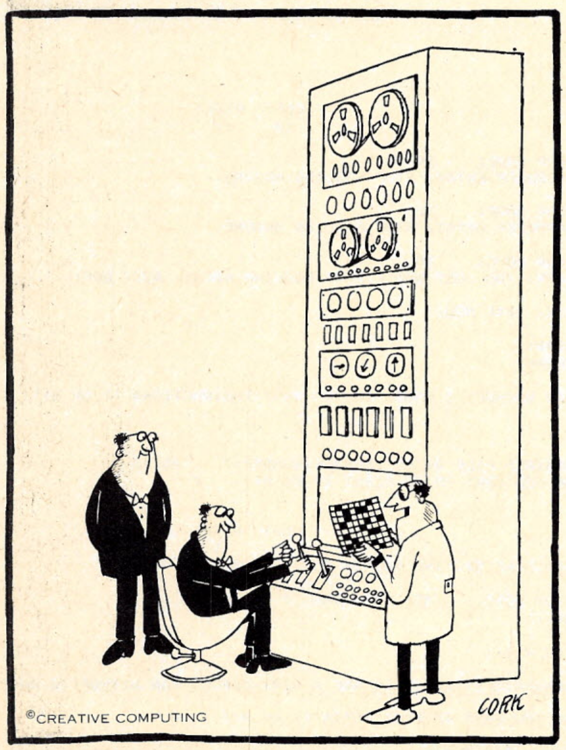
$P(B) = P(\mathrm{II}) = \dfrac{5 \cdot 6 \cdot 5}{6^5}$. In this case, $P(Y) = 1/6$ on roll 3.

$P(C) = P(\mathrm{III}) = \dfrac{10 \cdot 6 \cdot 25}{6^5}$. In this case, $P(Y) = (1/6)^2$ on roll 3.

$P(D) = P(\mathrm{IV}) = \dfrac{15 \cdot 10 \cdot 3 \cdot 4}{6^5}$. In this case, $P(Y) = (1/6)^3$ on roll 3.

$P(E) = P(\mathrm{V}) = \dfrac{10 \cdot 6 \cdot 60}{6^5}$. In this case, $P(Y) = (1/6)^3$ on roll 3.

$P(F) = P(\mathrm{VI}) = \dfrac{6 \cdot 5 \cdot 4 \cdot 3 \cdot 2}{6^5}$. In this case, $P(Y) = (1/6)^4$ on roll 3.



"And now: an animal with eight letters..."

Some Yahtzee players in Case VI like to leave one die and try to obtain Yahtzee by rolling the other dice. This will lead to a different analysis than we have discussed; however, the probability of a Yahtzee can be shown to be the same under either strategy.

The probability of rolling a Yahtzee is computed in Table I.

TABLE I

	roll 1		roll 2	roll 3		Decimal Product
I.	$1/6^4$		1	1		.00077
II.	$\dfrac{5 \cdot 6 \cdot 5}{6^5}$	A.	1/6	1		.00322
		B.	5/6	1/6		.00268
					Total:	.00590
III.	$\dfrac{10 \cdot 6 \cdot 25}{6^5}$	A.	$(1/6)^2$	1		.00536
		B.	$\dfrac{2 \cdot 5}{6^2}$	1/6		.00893
		C.	$(5/6)^2$	$(1/6)^2$		.00372
					Total:	.01801
IV.	$\dfrac{15 \cdot 10 \cdot 3 \cdot 4}{6^5}$	A.	$(1/6)^3$	1		.00107
		B.	$\dfrac{3 \cdot 5}{6^3}$	1/6		.00268
		C.	$\dfrac{3 \cdot 25}{6^3}$	$(1/6)^2$		.00223
		D.	$5/6^3$	$(1/6)^2$		.00015
		E.	$120/6^3$	$(1/6)^3$		.00060
					Total:	.00673
V.	$\dfrac{10 \cdot 6 \cdot 60}{6^5}$	A.	$(1/6)^3$	1		.00214
		B.	$\dfrac{3 \cdot 5}{6^3}$	1/6		.00536
		C.	$\dfrac{3 \cdot 25}{6^3}$	$(1/6)^2$		.00447
		D.	$5/6^3$	$(1/6)^2$		.00030
		E.	$120/6^3$	$(1/6)^3$		.00119
					Total:	.01346
VI.	$\dfrac{6 \cdot 5 \cdot 4 \cdot 3 \cdot 2}{6^5}$	A.	$1/6^4$	1		.00007
		B.	$150/6^5$	1/6		.00030
		C.	$1500/6^5$	$(1/6)^2$		.00050
		D.	$1800/6^5$	$(1/6)^3$		.00010
		E.	$3600/6^5$	$(1/6)^3$		.00020
		F.	$720/6^5$	$(1/6)^4$		.00001
					Total:	.00118

Thus, the probability of rolling a Yahtzee is: .00077 + .00590 + .01801 + .00673 + .01346 + .00118 = .04605. An experienced Yahtzee player knows very well how difficult it is to roll a Yahtzee. This probability thus supports his experience.

The following diagram displays an interesting set of partial probabilities. The probabilities for rolls 2 and 3 are assuming the prior result of roll 1.

ROLL 1		ROLLS 2 and 3
P = .00077 Yahtzee		
P = .01929 4 of a kind	→	P = .30556 Yahtzee
P = .19290 3 of a kind	→	P = .09336 Yahtzee
P = .23148 2 pairs	→	P = .02906 Yahtzee
P = .46296 1 pair	→	P = .02906 Yahtzee
P = .09259 No matches	→	P = .01263 Yahtzee

The interested reader and his students are invited to consult the rules for playing Yahtzee and compute the probabilities of other desirable outcomes. The need for a calculating device is apparent. ■

VAN GAM

Alan Brown*

VAN GAM is a simple game with an interesting solution set. The winning sequence pairs are formed by certain mutually exclusive *sequences,* using the golden mean, $\frac{1+\sqrt{5}}{2}$ as an irrational generator. See explanation, lines 40-130.

$$\text{IF } T = \frac{1+\sqrt{5}}{2}$$

$$\text{and } X = T + 1,\ Y = \frac{1}{T} + 1$$

then for integers N the winning sequence generators are

INT (N * X) 2 5 7 10 13 . . . .

INT (N * Y) 1 3 4 6 8 . . . .

It is interesting to note that the union of these sequences is the set of integers, and their intersection is empty. That is the case, in fact, for *any* irrational generator, but only T will produce winning VAN GAM pairs. The game is not much fun in that the average user will never be able to beat the computer in non-trivial cases, unless he has been taught the winning sequences.

*Headmaster, St. George's School, 4175 West 29 Ave., Vancouver, B.C., Canada v6S 1V6.

Sample Run

```
VAN WYTHOFF'S GAME: DO YOU WANT INSTRUCTIONS (Y OR N)?    Y
YOU ARE TO CREATE TWO PILES OF MATCHES, EACH CONTAINING 100 OR LESS.
YOU PLAY ALTERNATELY WITH ME, AND OUR MOVES CONSIST OF:
        (A) TAKING AWAY 1 OR MORE MATCHES FROM ONE PILE ONLY, OR
        (B) TAKING AWAY THE SAME NUMBER FROM EACH PILE.
THE ONE WHO TAKES AWAY THE LAST MATCH OF ALL WINS.
ENTER YOUR MOVES IN THIS MANNER:
             2L - (2 LEFT) TAKE TWO FROM LEFT PILE
             3R - (3 RIGHT) TAKE THREE FROM RIGHT PILE
             5B - (5 BOTH) TAKE FIVE FROM EACH PILE



DESIRED PILE SIZES (NUMBER,NUMBER)?    17,22
DO YOU WANT TO GO FIRST (Y OR N)?    Y


                           LEFT   RIGHT
                            17     22

YOUR MOVE:     3L    LEAVING  14     22
HM.. I TAKE:   2B    LEAVING  12     20

YOUR MOVE:     3B    LEAVING  9      17
HM.. I TAKE:   2R    LEAVING  9      15

YOUR MOVE:     5R    LEAVING  9      10
HM.. I TAKE:   3L    LEAVING  6      10

YOUR MOVE:     3B    LEAVING  3      7
HM.. I TAKE:   2R    LEAVING  3      5

YOUR MOVE:     3R    LEAVING  3      2
HM.. I TAKE:   2L    LEAVING  1      2

YOUR MOVE:     1R    LEAVING  1      1
HM.. I TAKE:   1B    LEAVING  0      0

SORRY - I WIN. DON'T FEEL BADLY - I'M AN EXPERT.

DO YOU WANT TO PLAY AGAIN (Y OR N)?    Y


DESIRED PILE SIZES (NUMBER,NUMBER)?    26,16
DO YOU WANT TO GO FIRST (Y OR N)?    N


                           LEFT   RIGHT
                            26     16
I TAKE:        1L    LEAVING  25     16

YOUR MOVE:     2B    LEAVING  23     14
HM.. I TAKE:   1L    LEAVING  22     14

YOUR MOVE:     2B    LEAVING  20     12
HM.. I TAKE:   1L    LEAVING  19     12

YOUR MOVE:     1B    LEAVING  18     11
HM.. I TAKE:   1L    LEAVING  17     11

YOUR MOVE:     2B    LEAVING  15     9
HM.. I TAKE:   1L    LEAVING  14     9

YOUR MOVE:     1B    LEAVING  13     8
HM.. I TAKE:   1L    LEAVING  12     8

YOUR MOVE:     2B    LEAVING  10     6
HM.. I TAKE:   1L    LEAVING  9      6

YOUR MOVE:     2B    LEAVING  7      4
HM.. I TAKE:   1L    LEAVING  6      4

YOUR MOVE:     1B    LEAVING  5      3
HM.. I TAKE:   1L    LEAVING  4      3

YOUR MOVE:     2B    LEAVING  2      1
HM.. I TAKE:   1L    LEAVING  1      1

YOUR MOVE:     1B    LEAVING  0      0
HM..

YOU WIN!!

CONGRATULATIONS. YOU ARE A VERY CLEVER VAN WYTHOFF'S GAMESMAN.

DO YOU WANT TO PLAY AGAIN (Y OR N)?    Y


DESIRED PILE SIZES (NUMBER,NUMBER)?    15,17
DO YOU WANT TO GO FIRST (Y OR N)?    Y


                           LEFT   RIGHT
                            15     17

YOUR MOVE:     19
IMPROPER ENTRY. STOP FOOLING AROUND.

YOUR MOVE:     22
IMPROPER ENTRY. STOP FOOLING AROUND.

YOUR MOVE:     18
LOOK, YOU JUST WON'T STOP FOOLING ABOUT. BUZZ OFF.

O.K.  BYE NOW.

READY
RUNNH

VAN WYTHOFF'S GAME: DO YOU WANT INSTRUCTIONS (Y OR N)?    N


DESIRED PILE SIZES (NUMBER,NUMBER)?    45,45
DO YOU WANT TO GO FIRST (Y OR N)?    Y


                           LEFT   RIGHT
                            45     45
YOU LIKE SITTING DUCKS, DON'T YOU?

YOUR MOVE:     45B   LEAVING  0      0
HM..

YOU WIN!!

CONGRATULATIONS. YOU ARE A VERY CLEVER VAN WYTHOFF'S GAMESMAN.

DO YOU WANT TO PLAY AGAIN (Y OR N)?    N

O.K.  BYE NOW.

READY
```

Program Listing

```
10 DIM Q(200)
15 X=(1+SQR(5))/2 \ Y=1+1/X \ X=1+X
16 FOR I=0 TO 99
17 Q(I*2)=INT(I*X) \ Q(I*2+1)=INT(I*Y)
18 NEXT I \ T=0
20 PRINT "VAN WYTHOFF'S GAME: DO YOU WANT INSTRUCTIONS (Y OR N)?    ";
30 A=SYS(4) \ PRINT  \ IF A=78 THEN 130
40 PRINT "YOU ARE TO CREATE TWO PILES OF MATCHES, EACH CONTAINING 100 OR LESS."
50 PRINT "YOU PLAY ALTERNATELY WITH ME, AND OUR MOVES CONSIST OF:"
60 PRINT TAB(10);"(A) TAKING AWAY 1 OR MORE MATCHES FROM ONE PILE ONLY, OR"
70 PRINT TAB(10);"(B) TAKING AWAY THE SAME NUMBER FROM EACH PILE."
80 PRINT "THE ONE WHO TAKES AWAY THE LAST MATCH OF ALL WINS."
90 PRINT "ENTER YOUR MOVES IN THIS MANNER:"
100 PRINT TAB(10);"2L - (2 LEFT) TAKE TWO FROM LEFT PILE"
110 PRINT TAB(10);"3R - (3 RIGHT) TAKE THREE FROM RIGHT PILE"
120 PRINT TAB(10);"5B - (5 BOTH) TAKE FIVE FROM EACH PILE"
130 PRINT  \ PRINT  \ PRINT
200 PRINT "DESIRED PILE SIZES (NUMBER,NUMBER)?    ";
205 A=SYS(3)
210 GOSUB 350 \ IF S1=0 THEN 210 \ L=S1 \ PRINT ",";
220 GOSUB 350 \ IF S1=0 THEN 220 \ R=S1 \ A=SYS(2)
230 IF L+R=0 THEN 250 \ IF L+R>5 THEN 240 \ PRINT  \ PRINT "OH, YOU'RE A SPORT, YOU ARE."
240 PRINT  \ PRINT "DO YOU WANT TO GO FIRST (Y OR N)?    ";
250 B=SYS(4) \ PRINT  \ PRINT  \ PRINT TAB(27);"LEFT  RIGHT"
255 PRINT TAB(27);L;TAB(33);R
260 P=L*2 \ IF L>R THEN 400 \ P=R*2 \ GO TO 400
350 S1=0
355 A=SYS(4) \ IF A>57 THEN 380 \ IF A<48 THEN 380
370 P1=10*S1+A-48 \ IF P1>100 THEN 380 \ S1=P1 \ PRINT CHR$(A); \ GO TO 355
380 RETURN
400 IF B=78 THEN 690
405 IF L=R THEN PRINT "YOU LIKE SITTING DUCKS, DON'T YOU?"
410 PRINT  \ PRINT "YOUR MOVE:";TAB(14); \ S1=0
430 A=SYS(3) \ A=SYS(4) \ IF A>57 THEN 480 \ IF A<48 THEN 480
440 P1=S1*10+A-48 \ IF P1>100 THEN 430 \ S1=P1 \ PRINT CHR$(A); \ GO TO 430
480 IF S1=0 THEN 430 \ IF A=76 THEN 550 \ IF A=82 THEN 570 \ IF A=66 THEN 590
490 GO TO 430
520 T=T+1 \ IF T>2 THEN 1940
530 PRINT  \ PRINT "IMPROPER ENTRY. STOP FOOLING AROUND." \ GO TO 410
550 IF S1>L THEN 520 \ L=L-S1 \ GO TO 610
570 IF S1>R THEN 520 \ R=R-S1 \ GO TO 610
590 IF S1>L THEN 520 \ L=L-S1 \ GO TO 570
610 PRINT CHR$(A);TAB(19);"LEAVING";TAB(27);L;TAB(33);R \ A=SYS(2)
614 PRINT "HM.. ";
615 FOR I=1 TO 500 \ A=A+I \ NEXT I
620 IF L+R>0 THEN 690
630 PRINT  \ PRINT  \ PRINT "YOU WIN!!" \ PRINT
635 PRINT "CONGRATULATIONS. YOU ARE A VERY CLEVER VAN WYTHOFF'S GAMESMAN."
640 GO TO 1900
690 I=0 \ M=0
700 IF Q(I)=L THEN 760 \ IF Q(I+1)=L THEN 780
720 IF Q(I)=R THEN 800 \ IF Q(I+1)=R THEN 820
740 I=I+2 \ GO TO 700
760 L1=I \ L2=1 \ IF M=1 THEN 840
770 M=1 \ GO TO 720
780 L1=I \ L2=0 \ IF M=1 THEN 840
790 M=1 \ GO TO 720
800 R1=I \ R2=1 \ IF M=1 THEN 840
810 M=1 \ GO TO 740
820 R1=I \ R2=0 \ IF M=1 THEN 840
830 M=1 \ GO TO 740
840 IF L=R THEN 895 \ IF R1=L1 THEN 1080
850 IF L1>R1 THEN 900
860 P=L1+L2
880 IF Q(P)>R THEN 940
890 M=R-Q(P) \ R=Q(P) \ A$="R" \ GO TO 1110
895 M=L \ L=0 \ R=0 \ A$="B" \ GO TO 1110
900 P=R1+R2
920 IF Q(P)>L THEN 940
930 M=L-Q(P) \ L=Q(P) \ A$="L" \ GO TO 1110
940 M=0 \ A$="B"
950 P=L1 \ IF R1<L1 THEN P=R1
960 P=P-2 \ G=0
965 M=M+1 \ L=L-1 \ R=R-1
970 FOR I=P TO 0 STEP -2
990 IF Q(I)=L THEN 1040
1000 IF Q(I+1)=L THEN 1050
1010 IF Q(I)=R THEN 1060
1020 IF Q(I+1)=R THEN 1070
1030 NEXT I \ GO TO 965
1040 IF Q(I+1)=R THEN 1110 \ GO TO 1075
1050 IF Q(I)=R THEN 1110 \ GO TO 1075
1060 IF Q(I+1)=L THEN 1110 \ GO TO 1075
1070 IF Q(I)=L THEN 1110
1075 IF G=1 THEN 960 \ G=1 \ GO TO 965
1080 IF L>R THEN 1100
1090 R=R-1 \ M=1 \ A$="R" \ GO TO 1110
1100 L=L-1 \ M=1 \ A$="L"
1110 PRINT "I TAKE:";TAB(14);STR$(M);A$;TAB(19);"LEAVING";TAB(27);L;TAB(33);R
1120 IF L+R>0 THEN 410
1130 PRINT  \ PRINT "SORRY - I WIN. DON'T FEEL BADLY - I'M AN EXPERT."
1900 PRINT  \ PRINT "DO YOU WANT TO PLAY AGAIN (Y OR N)?    ";
1910 A=SYS(4) \ PRINT
1920 IF A=89 THEN 130 \ GO TO 1950
1940 PRINT  \ PRINT "LOOK, YOU JUST WON'T STOP FOOLING ABOUT. BUZZ OFF."
1945 A=SYS(2)
1950 PRINT  \ PRINT "O.K.  BYE NOW."
1960 END
```

KIRKOF and NETWRK

Alan Brown*

KIRKOF inputs data describing a simple electrical network (resistors and DC potential difference sources). The program calculates the current in each line, using Kirchoff's Laws, and then chains (line 4300) to NETWRK which draws a picture of the circuit. Each program runs in 4K in the present configuration, which allows for 10 junctions (each labelled with a single digit or alphabetic character) and 22 possible current-bearing lines.

We run on a PDP/11V03 in a multi-user BASIC environment. There are several explanations required for some programming steps with this interpreter:

1. Note the role of backslashes in IF statement THEN statement statements. Line 510 in KIRKOF has the following effect: IF P (J,1) = I THEN (A(I,J) = 1 GO TO 550). An IF statement THEN line number treats backlashes differently; for example, line 1050 has no such parenthesized effect.

2. SEG$, CHR$, STR$ and TRM$ are particular string features of the PDP BASIC 11 interpreter.

3. A = SYS(3) eliminates echoing, SYS(2) restores it, A = SYD(4) inputs into A the ASCII code for a keyboard entered character.

Programming comments:

To merge the two programs in an 8K environment, omit lines 1 and 4300, alter 4260, renumber 4310, 4320.

KIRKOF: Lines	20- 90	instructions
	110- 391	data input
	400- 493	data reorganization into canonical form
	500- 550	current balance equations
	790-2000	Kirchoff's Loops identified and P.D. equations formed
	4000-4290	Solution of N x N equations by Gauss-Seidel interative method.

NETWRK: Quite a tricky program in avoidance of unwanted vertical lines. A bit too ad hoc, I'm afraid.

*Headmaster, St. George's School, 4175 West 29 Ave., Vancouver, B.C., Canada V6S 1V6.

"Let's see now ... Have you had any previous experience with electricity?"

KIRKOF
Program Listing

```
1 COMMON P(22,5),Q(10),X(22),N,M
9 A=SYS(3)
10 DIM A(22,22),B(22),S(22,2),D(22)
20 PRINT "DO YOU WANT INSTRUCTIONS? ";
30 A=SYS(4) \ IF A=78 THEN PRINT " NO" \ GO TO 90
40 IF A<>89 THEN 30 \ PRINT " YES" \ PRINT "ENTER DATA THUS:"
42 PRINT
48 PRINT "         20 OHMS              DATA ENTRY:"
50 PRINT "     ._/\/\/\__.________.     A C R20"
52 PRINT "     |         |        |     (OR C A R20)"
54 PRINT "     |         |        |     B A R50"
56 PRINT "     |         \        |     B C R10"
58 PRINT "     |         /10 OHMS |     B C -25"
60 PRINT "    A|         \        |C    (OR C B +25)"
62 PRINT "     |         /        |"
64 PRINT "     |         |        |"
66 PRINT "     ._/\/\/\__.___I|___."
68 PRINT "       50 OHMS  B  25 V."
69 PRINT  \ PRINT "AFTER EACH ENTRY, HIT THE SPACE BAR. WHEN FINISHED ALL ENTRIES, HIT RETURN."
70 PRINT  \ PRINT
90 PRINT  \ PRINT "ENTER NETWORK CONNECTIONS:" \ PRINT
110 FOR I=1 TO 2
120 A=SYS(4) \ IF A=13 THEN IF N=0 THEN 120 \ IF I=1 THEN A=SYS(2) \ PRINT  \ GO TO 400
130 IF A>47 THEN IF A<58 THEN 145
135 IF A>64 THEN IF A<91 THEN 145
140 GO TO 120
145 IF I=2 THEN IF Q(P(N+1,1))=A THEN 120
150 PRINT CHR$(A);" "; \ FOR J=1 TO M \ IF Q(J)=A THEN 170 \ NEXT J \ M=M+1 \ Q(M)=A \ J=M
170 P(N+1,I)=J \ NEXT I \ N=N+1
180 A=SYS(4) \ IF A=43 THEN P(N,3)=1 \ GO TO 220
190 IF A=45 THEN P(N,3)=-1 \ GO TO 220
200 IF A<>82 THEN 180
220 PRINT "  ";CHR$(A); \ S=0
230 A=SYS(4) \ IF A>47 THEN IF A<58 THEN S=S*10+A-48 \ PRINT CHR$(A); \ GO TO 230
250 IF A<>46 THEN 320
300 PRINT "."; \ C=10
310 A=SYS(4) \ IF A>47 THEN IF A<58 THEN S=S+(A-48)/C \ C=C*10 \ PRINT CHR$(A); \ GO TO 310
320 IF A<>69 THEN 390 \ PRINT "E"; \ C=1
330 A=SYS(4) \ IF A=45 THEN C=C*-1 \ PRINT "-"; \ GO TO 330
340 IF A>47 THEN IF A<58 THEN S=S*10^((A-48)*C) \ PRINT CHR$(A); \ GO TO 390
350 GO TO 330
390 P(N,4)=S \ PRINT " |  "; \ H=H+1 \ IF H=5 THEN H=0 \ PRINT
391 GO TO 110
400 FOR I=2 TO N \ IF P(I,1)<P(I,2) THEN 450 \ T=P(I,1) \ P(I,1)=P(I,2) \ P(I,2)=T \ P(I,3)=P(I,3)*-1
450 NEXT I
460 FOR I=1 TO N-2 \ FOR J=2 TO N-I
470 IF P(J,1)<P(J+1,1) THEN 490
480 IF P(J,1)=P(J+1,1) THEN IF P(J,2)<=P(J+1,2) THEN 490
485 FOR K=1 TO 4 \ T=P(J,K) \ P(J,K)=P(J+1,K) \ P(J+1,K)=T \ NEXT K
490 NEXT J \ NEXT I
493 PRINT
500 FOR I=1 TO M-1 \ FOR J=1 TO N
510 IF P(J,1)=I THEN A(I,J)=1 \ GO TO 550
520 IF P(J,2)=I THEN A(I,J)=-1
550 NEXT J \ NEXT I
790 FOR G=M TO N
800 L=L+1 \ IF P(L,1)<P(L+1,1) THEN 800 \ S1=1 \ S(1,1)=L \ S(1,2)=2
900 S5=L
1000 K=P(S(S1,1),S(S1,2)) \ FOR I=S5+1 TO N \ IF I=S(S1,1) THEN 1040
1005 IF K=P(I,1) THEN K1=2 \ GO TO 1100
1010 IF K=P(I,2) THEN K1=1 \ GO TO 1100
1040 NEXT I
1050 IF S1=1 THEN 800 \ S5=S(S1,1) \ S1=S1-1 \ GO TO 1000
1100 FOR J=1 TO S1 \ IF P(S(J,1),S(J,2))=P(I,K1) THEN 1040 \ NEXT J
1110 S1=S1+1 \ S(S1,1)=I \ S(S1,2)=K1 \ IF P(I,K1)<>P(S(1,1),1) THEN 900
1120 FOR I=1 TO S1 \ I1=S(I,1) \ P9=-P(I1,4)*((2*S(I,2))-3) \ IF P(I1,3)=0 THEN A(G,I1)=P9 \ GO TO 1300
1130 B(G)=B(G)+P(I1,3)*P9
1300 NEXT I
2000 NEXT G
4000 FOR K=1 TO N-1 \ P=ABS(A(K,K)) \ I1=K
4020 FOR I=K+1 TO N \ Q=ABS(A(I,K)) \ IF Q>P THEN P=Q \ I1=I
4025 NEXT I
4030 IF P=0 THEN 4310 \ IF I1=K THEN 4060
4040 FOR J=K TO N \ T=A(K,J) \ A(K,J)=A(I1,J) \ A(I1,J)=T \ NEXT J
4050 T=B(K) \ B(K)=B(I1) \ B(I1)=T
4060 FOR I=K+1 TO N \ M3=-A(I,K)/A(K,K) \ A(I,K)=0
4080 FOR J=K+1 TO N \ A(I,J)=A(I,J)+A(K,J)*M3 \ NEXT J
4090 B(I)=B(I)+B(K)*M3 \ NEXT I \ NEXT K
4210 FOR I=1 TO N \ D(I)=B(I) \ FOR J=I TO N \ D(I)=D(I)-A(I,J)*X(J) \ NEXT J \ NEXT I
4250 FOR I=N TO 1 STEP -1 \ FOR J=I+1 TO N \ D(I)=D(I)-D(J)*A(I,J) \ NEXT J
4260 IF A(I,I)=0 THEN 4310 \ D(I)=D(I)/A(I,I) \ NEXT I
4290 P=0 \ FOR I=1 TO N \ X(I)=X(I)+D(I) \ P=P+D(I)*D(I) \ NEXT I \ IF P>1.00000E-03 THEN 4210
4300 CHAIN "NETWRK.B01"
4310 PRINT "SHORT CIRCUIT IN LINE ";
4320 PRINT CHR$(Q(P(K,1)));"-";CHR$(Q(P(K,2)))
9999 END
```

NETWORK
Program Listing

```
4301 DIM Q1(10),Q5(10),P$(3)
4304 PRINT  \ PRINT
4305 B1$=" " \ L1$="!" \ P1$="."
4307 PRINT "  LINE    CURRENT"
4310 FOR I=1 TO N \ PRINT STR$(I);P1$;B1$;CHR$(Q(P(I,1)));"-";
4311 PRINT CHR$(Q(P(I,2)));"  ";STR$(X(I));" A" \ NEXT I
4320 PRINT  \ PRINT
5000 B$="            " \ FOR I=1 TO M-1 \ PRINT CHR$(Q(I));B$; \ NEXT I \ PRINT CHR$(Q(M))
5005 FOR I=1 TO M-1 \ PRINT L1$;B$; \ NEXT I \ PRINT L1$
5008 FOR I=1 TO N \ Q5(P(I,1))=Q5(P(I,1))+1 \ Q5(P(I,2))=Q5(P(I,2))+1 \ NEXT I
5010 M0=1 \ N1=1
5030 FOR I=N1 TO N
5032 IF P(I,1)>=M0 THEN IF P(I,5)=0 THEN 5034
5033 GO TO 5040
5034 N1=I \ M1=P(N1,1) \ M2=P(N1,2) \ P(N1,5)=1
5035 Q5(M1)=Q5(M1)-1 \ Q5(M2)=Q5(M2)-1
5038 M7=M7+1 \ GO TO 5200
5040 NEXT I \ M9=M0 \ GOSUB 6200 \ FOR M9=M0+1 TO M \ FOR J=0 TO 3
5041 P$(J)=P$(J)+B$ \ NEXT J \ M8=1 \ GOSUB 6200 \ M8=0 \ NEXT M9
5050 PRINT TRM$(P$(0)) \ FOR J=0 TO 3 \ PRINT TRM$(P$(J)) \ P$(J)="" \ NEXT J \ IF M7=N THEN 9990 \ GO TO 5010
5200 M9=M0
5210 IF M0=M1 THEN 5380
5220 GOSUB 6200 \ FOR I=0 TO 3 \ P$(I)=P$(I)+B$ \ NEXT I
5310 FOR I=M0+1 TO M1-1 \ V$=L1$ \ IF Q1(I)=0 THEN V$=B1$
5320 FOR J=0 TO 3 \ P$(J)=P$(J)+V$+B$ \ NEXT J \ NEXT I \ M9=M1
5380 GOSUB 6200
5393 R$=STR$(X(N1)) \ IF LEN(R$)<10 THEN R$=R$+" A"
5394 IF X(N1)<0 THEN R$="<"+R$ \ GO TO 5400
5396 R$=R$+"->"
5400 V$=B$ \ IF LEN(R$)<13 THEN R$=B1$+R$
5410 R$=R$+SEG$(B$,1,13-LEN(R$)) \ IF M2-M1=1 THEN V$=R$
5420 P$(1)=P$(1)+V$ \ IF P(N1,3)=1 THEN V$="--:I----------" \ GO TO 5460
5440 IF P(N1,3)=-1 THEN V$="--I:----------" \ GO TO 5460
5450 V$="-'\/\/\/\/`--"
5460 P$(2)=P$(2)+V$ \ V$=STR$(P(N1,4)) \ IF P(N1,3)=0 THEN 5470
5465 IF LEN(V$)<12 THEN V$=V$+" V" \ GO TO 5478
5470 IF LEN(V$)<10 THEN V$=V$+" OHM" \ GO TO 5478
5475 V$=V$+" O"
5478 IF LEN(V$)<13 THEN V$=B1$+V$
5480 P$(3)=P$(3)+V$+SEG$(B$,1,13-LEN(V$))
5490 P$(0)=P$(0)+B$
5500 FOR I=M1+1 TO M2-1 \ V$=B1$ \ IF Q1(I)=1 THEN V$=L1$
5510 P$(0)=P$(0)+V$+B$ \ P$(1)=P$(1)+V$
5520 P$(3)=P$(3)+V$+B$ \ V$=B$ \ IF I=M2-1 THEN IF M2-M1>1 THEN V$=R$
5530 P$(1)=P$(1)+V$ \ P$(2)=P$(2)+"---------------"
5540 NEXT I \ M0=M2 \ GO TO 5030
6200 V$=B1$ \ IF Q1(M9)=1 THEN V$=L1$
6250 P$(0)=P$(0)+V$ \ P$(1)=P$(1)+V$
6252 V$=B1$ \ IF M8<>1 THEN 6255
6253 IF Q1(M9)=1 THEN V$=L1$
6254 GO TO 6257
6255 IF Q1(M9)+Q5(M9)>0 THEN V$=P1$
6256 IF M0=M1 THEN V$=P1$
6257 P$(2)=P$(2)+V$
6260 IF M8<>1 THEN Q1(M9)=0 \ IF Q5(M9)>0 THEN Q1(M9)=1
6265 V$=B1$ \ IF Q1(M9)=1 THEN V$=L1$
6270 P$(3)=P$(3)+V$
6280 RETURN
9990 PRINT  \ PRINT  \ PRINT  \ PRINT "ANOTHER NETWORK?   "; \ A=SYS(3) \ A=SYS(4)
9991 IF A<>89 THEN 9998 \ PRINT "YES" \ CHAIN "KIRKOF.B01" LINE 90
9998 PRINT "NO" \ A=SYS(2)
9999 END
```

Sample Run

```
DO YOU WANT INSTRUCTIONS?  NO

ENTER NETWORK CONNECTIONS:

A B   R110 !  B C   +100 !  C D   R120 !  E F   R130 !  A G   -150 !
B D   R140 !  E G   R150 !  B D   R160 !  E H   R170 !  B E   +200 !
B H   R180 !  C F   R190 !  E G   R200 !



  LINE    CURRENT
1. A-B  1.78832 A
2. A-G  -1.78832 A
3. B-C  .201199 A
4. B-D  -.273973 A
5. B-D  -.239726 A
6. B-E  2.67225 A
7. B-H  -.571428 A
8. C-D  .513699 A
9. C-F  -.3125 A
10. E-F  .3125 A
11. E-G  1.0219 A
12. E-G  .766423 A
13. E-H  .571429 A
```

```
A             B             C             D             E             F             G             H
!             !             !             !             !             !             !             !

  1.78832 A->   .201199 A->   .513699 A->                 .3125 A->
.-'\/\/\/\/`--.--!I---------.-'\/\/\/\/`--.             .-'\/\/\/\/`--.
! 110 OHM     ! 100 V       ! 120 OHM     !             ! 130 OHM     !
!             !             !             !             !             !
!             !             !             !             !             !
!             !             !             !             !             ! <-1.78832 A
.--I!-------------------------------------------------------------------------------.
  150 V       !             !             !             !             !             !
              !             !             !             !             !             !
              !             !             !             !             !             !
              !             ! <-.273973 A !             !             ! 1.0219 A->  !
              .-'\/\/\/\/`----------------.             .-'\/\/\/\/`----------------.
              ! 140 OHM     !             !             ! 150 OHM     !             !
              !             !             !             !             !             !
              !             !             !             !             !             !
              !             ! <-.239726 A !             !             ! .766423 A-> !
              .-'\/\/\/\/`----------------.             .-'\/\/\/\/`----------------.
              ! 160 OHM     !                           ! 200 OHM     !
              !             !                           !             !
              !             !                           !             !
              !             !               2.67225 A-> !             !               .571429 A->
              .--!I-------------------------------------.-'\/\/\/\/`------------------------------.
              ! 200 V       !                             170 OHM     !                           !
              !             !                                         !                           !
              !             !                                         !                           !
              !             !                                         !               <-.571428 A !
              .-'\/\/\/\/`------------------------------------------------------------------------.
                180 OHM     !                                         !
                            !                                         !
                            !                                         !
                            !                             <-.3125 A   !
                            .-'\/\/\/\/`------------------------------.
                              190 OHM
```

ANOTHER NETWORK?   YES

ENTER NETWORK CONNECTIONS:

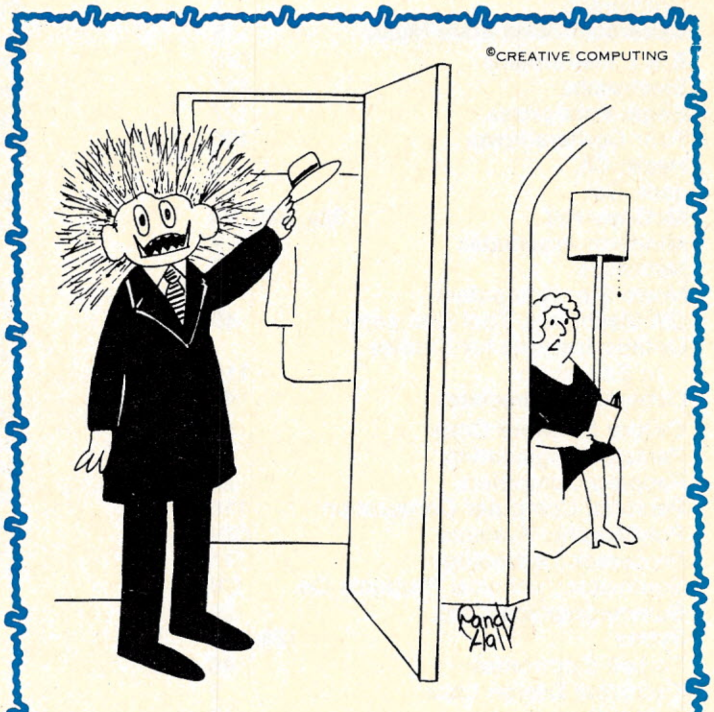
```
1 2   R15 !  1 3   R10 !  3 4   R25 !  4 2   R20 !  1 4   -50 !
2 3   +75 !
```

```
   LINE   CURRENT
1. 1-2  3.8961 A
2. 1-3  -1.65584 A
3. 1-4  -2.24026 A
4. 2-3  4.31818 A
5. 2-4  -.422078 A
6. 3-4  2.66234 A
```

```
1             2             3             4
!             !             !             !

  3.8961 A->    4.31818 A->   2.66234 A->
.-'\/\/\/\/`--.--!I---------.-'\/\/\/\/`--.
! 15 OHM      ! 75 V        ! 25 OHM      !
!             !             !             !
!             !             !             !
!             ! <-1.65584 A !             !
.-'\/\/\/\/`----------------.             !
! 10 OHM      !                           !
!             !                           !
!             !                           !
!             !               <-2.24026 A !
.--I!-------------------------------------.
  50 V        !                           !
              !                           !
              !                           !
              !               <-.422078 A !
              .-'\/\/\/\/`----------------.
                20 OHM
```

ANOTHER NETWORK?   NO

READY



"How did things go at the lab today Dear?"

Index to Advertisers

Coming in March

• **Small-Business Uses of Personal Computers.** In the first part of this new series, which will cover a specific area in each issue, we'll examine programs for merchandise inventory. First as viewed by an independent consultant, followed by descriptions of several such programs, written by the software houses, hardware manufacturers and computer stores offering them.

• **Three Computer-Music Systems.** Steve North, our resident software genius, has tried out three highly different systems for producing music with a microcomputer. Read his penetrating analysis and see if one of the three is for you.

• **Using a Light Pen to Interact with Color Graphics.** Tom Dwyer, whose four-part series on BASIC (which ends in this issue) has attracted wide attention, takes a look at the light pen, specifically as used with the Compucolor and its eight-color graphics.

• **Games, Games, Games.** Several new ones you'll want to feed into your computer right away. Complete listings, runs and descriptions, naturally.

• **Can a Computer Really Play Winning Chess?** Our publisher, Dave Ahl, has looked into computerized chess, and reports on past history, current progress, and some of the available hardware and software for playing the "game of kings."

• **What the Beginning Computer Hobbyist Needs to Know.** Steve North, wearing his tutorial hat this time, answers such questions as: Which microcomputer is best for you? Which features should you base your decision on? How can you act smart and ask the right questions when you walk into your local computer store? — and many more. For a ground-floor introduction to micros and what makes them tick, don't miss these pages.

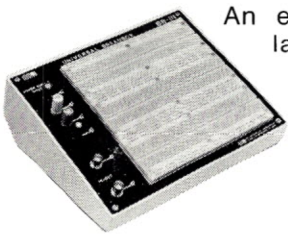
Solder is for sissies.

Are you still designing circuits the old fashioned way? Soldering and unsoldering? Burning up valuable IC's? If so, it's time you made a switch to solderless breadboarding. E&L Instruments offers a complete line of breadboarding products from sockets to the most complete systems for professional use. Here are five ways you can go solderless . . .

SK-10 Socket

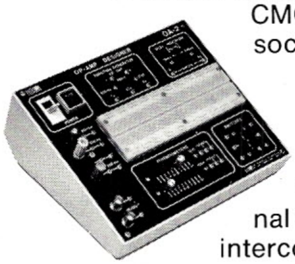
This precision solderless breadboarding socket will pay for itself in one afternoon. Integrated circuits and discrete components insert directly, without adapters. All interconnections are made with 20 to 26 gage solid conductor wire. $17.75.

Breadbox III



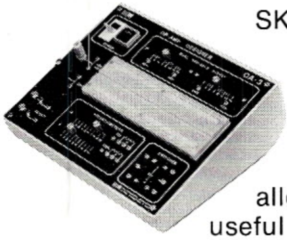
An economical approach to a large breadboarding area. Three SK-10 sockets, four 5-way binding posts, two BNC connectors, six BP-22 solderless breadboarding pins. $72.25

Op-Amp Designer #OA-2K



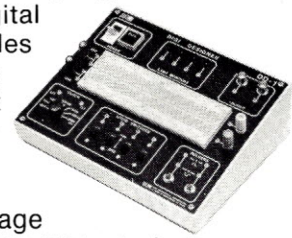
Powerful design tool for linear work and CMOS digital circuitry. SK-10 socket. Heavy current capacities on all three power supplies [+/− 15V & +5V DC]. Sensitive null detector and a full function generator with external frequency adjustment. All interconnections are solderless. $120.00 in kit form.

Op-Amp Designer #OA-3K



SK-10 socket. Plus and minus variable supplies (1 to 15V). This adjustable feature makes the unit exceptionally flexible, allowing it to be more useful for odd voltage requirements. The fixed 5 volt supply allows interfacing with TTL circuitry. $102.50 in kit form.

Digi-Designer #DD-1K



A must for every student, engineer or technician doing digital circuit design. Handles both IC and discrete components without soldering. Connections to the SK-10 socket are made using any solid 22 gage insulated wire. The DD-1K includes a clock, power supply, logic lights, pulsers, logic level switches and much more. All circuitry is now in one circuit board. $77.75 in kit form.

OEM and educational discounts available. All prices shown are suggested resale U.S.A.

E&L INSTRUMENTS, INC.
61 First Street, Derby, Conn. 06418
(203) 735-8774 Telex No. 96 3536

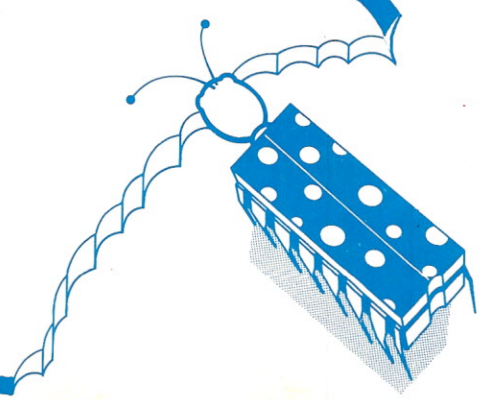
Authorized Stocking Representatives

Los Angeles, Calif.
(213) 377-0975
San Francisco, Calif.
(415) 961-2828
Woodbridge, Conn.
(203) 397-1461
Denver, Colorado
(303) 744-2501
Orlando, Florida
(305) 351-1841
Chicago, Illinois
(312) 956-8090
Kansas City, Kansas
(913) 649-8952
New York, N.Y.
(201) 467-8585
Syracuse, N.Y.
(315) 699-2651

Dayton, Ohio
(513) 222-0011
Philadelphia, Pa.
(215) 723-8733
Dallas, Texas
(214) 328-5484
Fairfax, Virginia
(703) 273-1803
Seattle, Washington
(206) 938-4166

Canada
Edmonton, Alberta
(403) 455-4122
Winnipeg, Manitoba
(204) 774-6286
or 772-9295

Vancouver,
British Columbia
(604) 687-2621



The Computer for the Professional

Whether you are a manager, scientist, educator, lawyer, accountant or medical professional, the System 8813 will make you more productive in your profession. It can keep track of your receivables, project future sales, evaluate investment opportunities, or collect data in the laboratory.

Use the System 8813 to develop reports, analyze and store lists and schedules, or to teach others about computers. It is easily used by novices and experts alike.

Reliable hardware and sophisticated software make this system a useful tool. Several software packages are included with the machine: an advanced disk operating system supporting a powerful BASIC language interpreter, easy to use text editor, assembler and other system utilities. Prices for complete systems start at $3250.

See it at your local computer store or contact us at 460 Ward Dr., Santa Barbara, CA 93111, (805) 967-0468.

PolyMorphic
Systems